# Modelling and Verifying Abilities of Rational Agents

**Doctoral Thesis**
**(Dissertation)**

to be awarded the degree of

**Doctor rerum naturalium (Dr. rer. nat.)**

submitted by

**Nils Bulling**
from Hamburg, Germany

approved by the

Faculty of Mathematics/Computer Science and Mechanical Engineering
Clausthal University of Technology

Date of oral examination
22.10.2010

Chairperson of the Board of Examiners
Prof. Dr. Jörg Müller

Chief Reviewer
Prof. Dr. Jürgen Dix

Reviewer
PD Dr. habil. Wojciech Jamroga
Prof. Dr. Sarit Kraus

# Contents

# Preface

Four years have passed since the beginning of my PhD studies in Clausthal. I still have strong memories of the end of my graduate studies when I was offered a PhD position by the new head of the Computational Intelligence Group, Jürgen Dix. After some time I accepted, and no doubt it was the right decision. Working in and being part of Jürgen's group has been a great experience. He has taught me a lot about science, research, logics, teaching, complexity, life, and many other things. He also gave me the possibility to participate in many conferences, to travel and to collaborate with colleagues around the world. I am very grateful for this and happy to have had Jürgen as supervisor. Jürgen, thank you very much again for all your help and support!

Wojtek Jamroga is another person I owe a lot. Wojtek introduced me to the field of logics for multi-agent systems. I met Wojtek for the first time when he was visiting Clausthal to apply for a position. For this purpose, he gave a talk on a logic for reasoning about abilities of agents. Well, at first sight the topic did not impress me very much, but it did not take much time until I changed my mind. Wojtek supervised my diploma thesis on the very same logic, while I studied abroad at Durham University in the UK. This paved the way for many successful collaborations, fruitful discussions, and important and nice experiences. I have been very lucky to have had Wojtek as colleague and co-supervisor. Thank you for all this!

Of course, the whole atmosphere in our group has been an important factor for my work. Conversations and discussions with my (former) colleagues Michael Köster, Peter Novák, Tristan Behrens, and Yingqian Zhang have broadened my mind and the social environment made me feel comfortable. I could always rely on their support and advice. I am glad that you were next door!

A PhD is usually not done in complete isolation. So it was in my case. I would like to thank my colleagues and co-authors Berndt Müller, Carlos

Chesñevar, Jürgen, and Wojtek (work with them also found its way in this thesis), and Koen Hindriks and Mehdi Dastani. Thank you for the great collaboration. I have really enjoyed working with you!

Finally, I am grateful to Sarit Kraus for acting as external reviewer and to Jörg Müller for being the chairperson of the Board of Examiners.

Furthermore, I would like to thank my friends Fabian Kirchhoff and Nils Altmüller who have helped me, among many other things, to find typos and errors in my thesis. In particular Fabian has supported me in many different ways during the last years. I could always count on him!

Last and most important, I thank my parents Sylvia and Peter and my friends and family Ingeborg and Rolf. They *always*–since my birth–supported and helped me whenever I needed support or help.

Clausthal-Zellerfeld, Germany

*Nils Bulling*
October 2010

# 1

# Introduction

## Contents



In this thesis we analyse the abilities of agents that act, one way or another, rationally. We propose logic-based methods for modelling and reasoning about such agents. Each of the presented frameworks addresses a different aspect of modelling rational behaviour. Finally, we analyse the complexity of the corresponding model checking problems.

## 1.1 Introduction

Temporal logics for specifying and verifying reactive systems have a long tradition in computer science. In [Pnueli, 1977] Amir Pnueli proposed the *linear time temporal logic* **LTL**, a temporal logic to reason about linear time properties of infinite computations. Properties of the kind "the system never ends up in a deadlock" ($\Box \bigcirc \top$) or "the reactor will eventually shut down" ($\Diamond \mathsf{shut\_down}$) can be expressed within this logic. Some years later, Edmund Clarke, Allen Emerson, and Joseph Halpern [Clarke and Emerson, 1981; Emerson and Halpern, 1982] proposed the *computation-tree logics* **CTL** and **CTL**$^\star$. These logics allow to quantify over different system behaviours; they model branching time rather than linear time. The logic **CTL** is a restricted variant of **CTL**$^\star$ with better computational properties. These logics became the most popular ones in computer science and are often used for specifying and verifying reactive systems [Clarke et al., 1986; Queille and Sifakis, 1982].

### 1.1.1 Agents and Multi-Agent Systems

The temporal logics **LTL**, **CTL**$^\star$, and their derivatives allow to reason about (temporal) properties (e.g. fairness, liveness, and safety properties) of reactive systems. At the end of the 20th century the notion of agent was introduced. The *agent paradigm* [Wooldridge, 2000, 2002; Weiss, 1999] provides a neat way to talk about (distributed) systems which are ascribed additional properties e.g. heterogeneity, autonomy, pro-activeness, rationality, intelligence, etc. Agents can be considered as (autonomous, etc.) processes and multi-process systems as multi-agent systems (MASs). Due to the specific settings and systems of interest, cooperation and group ability became focal points in research on MASs. Formal tools developed for reactive systems turned often out insufficient for the modelling and verification of specific properties (e.g. abilities of groups of agents) related to MASs. Suitable tools were required to address the important aspects of cooperation and power.

*Alternating time temporal logics* (ATLs) were proposed [Alur et al., 1997, 2002] to model open systems and the power of groups of agents. Nowadays they are among the most popular and influential logics for modelling and reasoning about strategic abilities of agents.

The logics **ATL**$^*$ and **ATL** can be considered as multi-agent extensions of **CTL**$^\star$ and **CTL**, respectively. Path quantifiers are replaced by coalitional operators $\langle\langle A\rangle\rangle$ which allow to reason about the *group ability* of the team $A$ of agents. These operators allow for a finer-grained quantification over computation paths (not only universally or existentially). Basic game-theoretic notions underly the semantics of the ATLs.

In order for a formula $\langle\langle A\rangle\rangle\gamma$ being true group $A$ is supposed to have a winning strategy to enforce $\gamma$. The opponents can behave in arbitrary ways. Analogously to **LTL** and **CTL**$^\star$ being used for the specification and verification of reactive (closed systems), the new class of logics is suitbale for specifying and verifying multi-agent systems (open systems).

However, although extensively studied and applied to MAS, ATLs do not take into account basic characteristics of MASs that go beyond the pure execution of actions, e.g. agents' goals or other mental attitudes that influence agents' behaviours.

### 1.1.2 Agents That Act Rationally

ATLs make use of very basic game-theoretic concepts when it comes to the joint execution of actions. The use of such methods is very limited. However, agents are often ascribed other properties like *pro-activeness*, *rationality*, or *intelligence* which are not addressed in the ATLs.

Arguably, there is a need to reason about *rational agents*. In the last years the variety of products has been steadily increased and new application domains have arisen. Good examples are the family of *e*-products: e-auctions, e-commerce, e-learning, e-gaming, e-book-store, e-recommender-systems, e-learning-platforms, etc. A characteristic which most of these products share is the interaction with humans. Programs try, e.g., to support, to influence, or to inform buyers and sellers. These are scenarios were reasoning about and the analysis of rational behaviour can pay off. A good mechanism supports the auctioneer to maximise the money he gets from the participators. The e-store increases its sales if it suggests products that match with the customers' interests, etc. Of course, humans do often *not* act rationally. Hence, in the following we consider rationality from a more abstractly as *plausible* behaviour with respect to a *certain context*.

Game theory and the analysis of rational behaviour have a long tradition. For example, in order to solve games it is usually assumed that players play rationally (e.g. think about the method of backward induction). So unsurprisingly, it it did not take long until research on ATLs has included other ideas from game theory. Several researchers have considered the interplay between strategic ability and knowledge (e.g. [van der Hoek and Wooldridge, 2002; Jamroga and van der Hoek, 2004; van Otterloo et al., 2003]). Questions typically asked were of the following kind:

- Are agents aware that they can win?
- If agents know that they can win do they also know *how*?

Another focus has been the characterisation and usage of game-theoretic solution concepts. The idea has been inspired by the way in which games are analysed in game theory. Firstly, game theory identifies a number of solution concepts (e.g., Nash equilibrium, undominated strategies, Pareto optimality) that can be used to define rational behaviour of players. Secondly, it is usually assumed that players play rationally in the sense of one of the above concepts, and it is asked about the outcome of the game under this assumption. The first issue has been studied in the framework of logic, for example in [Bacharach, 1987; Bonanno, 1991; Stalnaker, 1994, 1996]; more recently, game-theoretical solution concepts have been characterised in dynamic logic [Harrenstein et al., 2002, 2003], dynamic epistemic logic [Baltag, 2002; van Benthem, 2003], and extensions of **ATL** [van der Hoek et al., 2005a; Jamroga et al., 2005]. The second thread seems to have been neglected in logic-based research: The work [van Otterloo et al., 2004; van der Hoek et al., 2004; van Otterloo and Roy, 2005; van Otterloo and Jonker, 2004] are the only exceptions we know of. Moreover, each proposal from [van Otterloo et al., 2004; van der Hoek et al., 2004; van Otterloo and Roy, 2005; van Otterloo and Jonker, 2004] commits to a particular view of rationality (Nash equilibria, undominated strategies etc.).

This is one aspect to rational agents we try to generalise in this thesis. We propose a logic, *alternating time temporal logic with plausibility* (**ATLP**), that allows to "plug in" *any* solution concept of choice (that we are able to formalise). Moreover, we show that this logic does also allow to describe solution concepts in a more expressive way. The main idea is that we often know that agents behave according to some rationality assumptions, they are not completely dumb. Therefore we do not have to check *all possible plays* (what is the case for ATLs) – only those that are *plausible* in some reasonable sense.

Rationality means more than selecting appropriate strategies. Coalitions itself should have some underlying rationale in order to work together. In order to join a coalition, agents usually require some kind of *incentive* (e.g. sharing common goals, getting rewards, etc.), since usually forming a coalition does not come for free (fees have to be paid, communication costs may occur, etc.). To address this important aspect we extend ATLs by another dimension which takes the coalition formation process into account. For this purpose we combine an argumentative approach to coalition formation with the logics' semantics.

Both extensions are presented in the context of perfect information. Agents know the current state of the world and the opponents are able to communicate and cooperate arbitrarily. These assumptions do not always hold in MASs. For this purpose, we propose two logics to model such settings. For the first point we combine our method of plausibility used in **ATLP** with an epistemic logic for strategic ability. The proposed logic **CSLP** goes beyond the pure union of the parts it is composed of. The rationality concept allows us to neatly define the relationship between epistemic and doxastic concepts. Secondly, it enables us to analyse rational play under incomplete information and to describe appropriate solution concepts.

Another angle to incomplete information is addressed by adding probabilistic concepts to **ATL**. These concepts are used to *probabilistically predict* the opponents' behaviour if their ability to communicate and to find the "best" strategy is somehow limited. The motivation is that there are scenarios in which the common assumption that opponents behave in the most destructive way is not sensible. For example, due to the lack of communication channels agents may not be able to agree on their most destructive counter strategy. This is the case for multiple Nash equilibria. In order to play a Nash equilibrium strategy agents are required to signalise which of them they are going to select. We model this by assuming some *probabilistic behaviour* of the opponents.

Finally, we consider yet another angle to rationality in MASs. Up to now the viewpoint of agents being entities perceiving changes in their environment and acting according to a set of rules or plans in the pursuit of goals does not take resources into account. This assumption is not always realistic; especially in times of shared and globally offered applications the use of them does often not come for free. In order to be able to model such scenarios we assign costs to actions and show that the verification problem may become difficult. We show that it is usually impossible (i.e. undecidable) to reason about such settings without imposing strong restrictions. In parallel to our work, such settings were also considered by other researchers [Alechina et al., 2009b,a, 2010].

### 1.1.3 Verification of Rational Agents

We have motivated the use of logics for modelling rational agents. Apart form modelling, an automatic way to prove properties and to verify systems is of great importance. Verification of (multi-)agent systems is among the most important applications of linear time and strategic logics in computer science. Above, we have pointed out that **LTL** and **CTL**$^\star$ can be used for the specification of reactive systems. In particular, the model checking problem has attracted much attention. Model checking is the process to check whether a given formula holds in a given model. It is used to check whether a system complies with a given set of specifications. The complexity of this problem is proven to be **P**-complete (resp. **PSPACE**-complete) for **CTL** (resp. **LTL** and **CTL**$^\star$). Surprisingly, the multi-agent case for the basic logic **ATL** is not harder. However, for **ATL**$^*$ it is already **2EXPTIME**-complete. In the multi-agent cases the different settings of perfect information, imperfect information, perfect recall, and imperfect recall have an enormous influence on the complexity. For example, the case for **ATL**$^*$ with perfect recall and imperfect information is believed to be undecidable [Alur et al., 2002]. Clearly, this limits its practical use as a verification language for MASs.

Thus, in this thesis we do also consider the model checking problems for our proposed logics. We show that the picture is manifold. In general, the complexity of verifying rational agents is harder than for the "standard" cases (i.e. without rationality assumptions). However, we also have positive results that rationality must not necessarily increase the complexity of model checking. We show that resources make the verification particularly difficult, even undecidable in many settings.

## 1.2 Structure of The Thesis

The thesis is divided into the following five parts.

*Part I – Preliminaries*

In the first part background material and related work is presented. In Chapter 2 the basic temporal (e.g. **LTL**, **CTL**, and **CTL**$^*$) and strategic logics (e.g. **ATL** and **ATL**$^*$) are introduced. The differences and dependencies between perfect vs. imperfect recall and perfect vs. imperfect information are discussed.

Chapter 3 focusses on the connection between games and strategic logics. A brief introduction to game theory followed by a discussion on logics for reasoning about games is given. We also discuss logical characterisations of solution concepts.

Chapter 4 serves as a collection of theoretical background material needed in the thesis. Firstly, an introduction to complexity theory is given. Then, we present an argumentative approach to coalition formation and provide brief introductions to probability theory and to Petri nets. Chapter 5 is on model checking strategic logics. The general problem is formulated and upper and lower complexity bounds of the well-known temporal and strategic logics introduced in Chapter 2 are summarised.

*Part II – Rational Agents: Models and Logics*

This is one of the two main parts of this thesis. Models and logics for reasoning about rational agents are presented. In Chapter 6 we propose logics for rational agents with *perfect information.* The logic *alternating time temporal logic with plausibility* (**ATLP**) is introduced and it is shown how it can be used to characterise game-theoretic solution concepts and how to impose them on the agents in order to restrict their behaviour. Secondly, **ATL** is enriched by an argumentative approach to coalition formation to reason about the abilities of *rational coalitions.* The logic is called **CoalATL**.

In Chapter 7 we consider rational agents under *incomplete information.* The logic **CSLP** (*constructive strategic logic with plausibility*) is an incomplete information extension of **ATLP**. It allows to characterise and to speak about imperfect information games. Moreover, the concepts of knowledge and plausibility allow to define a neat notion of belief. The relations among these concepts are discussed. We also address another angle to incomplete information. The *alternating time temporal logic with probabilistic success* (**pATL**) allows to model abilities of agents under the assumption that the communication between the opponents is somehow restricted. It cannot be assumed – as it is done in the ATLs– that the opponents act in the most destructive way.

In the last Chapter of this part we consider *resource-bounded agents.* Firstly, we present resource-bounded logics for the single agent case (**RTL** and **RTL**$^\star$). Then, we turn to the more interesting case of multiple agents. We consider a flexible setting resulting in many different logics.

*Part III – Complexity of Verifying Rational Agents*

In Part III we consider the complexity of model checking the logics introduced in Part II. In Chapter 9 we analyse the agents with memory affect the model checking complexity. In Chapter 10 we analyse the logics for rational play. Then, in Chapter 11 we consider resources and show that these settings are generally much harder. Apart from a single-agent case and some restricted settings we show that the model checking problems are undecidable.

*Part IV and V – Conclusion and Appendix*

In Part IV we conclude and summarise related work. The last part contains detailed proofs and some additional material.

## 1.3 Publications

Some results reported in this thesis have been already published. In the following we list the publications relevant for each chapter.

- Chapter 2 includes a part from [Bulling et al., 2010].
- Chapter 3 takes chapters from [Bulling et al., 2009b].
- Chapter 5 is based on [Bulling et al., 2010] and [Bulling and Jamroga, 2010a, 2009a, 2010b].
- Chapter 6 incorporates several publications. The sections on **ATLP** appeared in [Bulling et al., 2009b] which in turn is based on [Bulling and Jamroga, 2007b; Jamroga and Bulling, 2007b,a]. The sections related to **CoalATL** are taken from [Bulling and Dix, 2010] and from [Bulling et al., 2008, 2009a; Bulling and Dix, 2008].
- Chapter 7 is a combination of work that appeared in [Bulling and Jamroga, 2009b] and [Bulling and Jamroga, 2009c, 2008].
- Chapters 8 and 11 are based on [Bulling and Farwer, 2010a] and [Bulling and Farwer, 2010c].
- Parts of chapter 9 has appeared in [Bulling and Jamroga, 2010a, 2009a, 2010b].
- Chapters 10 is based on the same publications as Chapters 6 and 7.

## 1.4 Notation

In the following we summarise notation used throughout the thesis.

### 1.4.1 General Symbols

$\mathbb{A}\mathrm{gt}$ denotes a non-empty and finite set of agents. If not specified otherwise, we assume that $\mathbb{A}\mathrm{gt} = \{1, \ldots, k\}$. We often use $a, b, \ldots$ and $a_1, a_2, \ldots$ as placeholders for agents. We use $A$, $A'$, $B$,... to denote groups of agents.
$\Pi$ is a non-empty set of propositional symbols. Propositions are typeset as follows: $\mathsf{p}, \mathsf{q}, \mathsf{A\_proposition}, \ldots$. Usually, $\pi$ is used as labelling function.
$Q$ is a non-empty set of states. If not stated otherwise we assume that the set is finite. Usually, we use $q, q', q_1, q_2, \ldots$ as names for states and also as variables referring to them.

$\lambda, \lambda', \lambda_1, \lambda_2, \ldots$ are used to denote paths.
$s_a$ denotes a strategy for agent $a$
$s_A$ is a set $\{s_a \mid a \in A\}$ of strategies, one per agent in $A$.
$s_\emptyset$ denotes the empty strategy of the empty coalition.
$s_A|_B$ is defined as $\{s_a \mid a \in A \cap B\}$.
$s_A|_a$ is defined as $s_a \in \{s_a \mid a \in A \cap \{a\}\}$.
$\Sigma_A$ denotes the set of common strategies of $A$.

$\mathbb{N}$ denotes the set of integers $\{1, 2, \ldots\}$.
$\mathbb{N}_0$ $:= \mathbb{N} \cup \{0\}$
$X^+$ is the set of all finite and non-empty sequences of elements over $X$, for non-empty $X$.
$X^*$ $:= X^+ \cup \{\epsilon\}$ for non-empty $X$.
$X^\omega$ denotes all infinite sequences of elements over $X$ for non-empty $X$.
$X^{\leq\omega}$ $:= X^+ \cup X^\omega$ for non-empty $X$.

### 1.4.2 Acronyms and Fonts.

ACRONYM: For *acronyms* we use the font ANACRONYM.
$\mathcal{L}_{Language}$: Logical languages are denoted $\mathcal{L}_{Logicname}$
**Logicname**: Logics are denoted by **ID** and are considered as triple ($\mathcal{L}_{ID}, \models_{\mathbf{ID}}$ , Struc) consisting of a logical language $\mathcal{L}_{ID}$, a satisfaction relation, and a class Struc of models. More precisely, we consider the valid sentences over this triple. Subscripts may be omitted; as well as, the class of structures if clear from context.
Proposition: For *propositions* we use the font proposition.

# Part I

# Preliminaries

# 2

# Temporal and Strategic Logics

## Contents



In this section we introduce standard logics for reasoning about time and strategic ability. We present the *linear time logic* **LTL** [Pnueli, 1977], the *computation-tree logics* **CTL** and **CTL**$^\star$ [Emerson and Halpern, 1986], and the *alternating time temporal logics* **ATL** and **ATL**$^*$ [Alur et al., 2002, 1997, 1998b] as well as variants of them. We use the acronyms LTLs, CTLs, and ATLs, respectively, to refer to the logics belonging to the same "class".

For the ATLs we discuss the relations between perfect vs. imperfect information on one hand, and perfect vs. imperfect recall on the other, and we show how they give rise to different logics.

We introduce two more logics for reasoning about epistemic properties in a strategic context. These are the *constructive strategic logic* (CSL) **CSL** [Jamroga and Ågotnes, 2006, 2007] and the *alternating time temporal epistemic logic* **ATEL** [van der Hoek and Wooldridge, 2003].

*Remark 2.1 (Notation).* In the rest of this thesis we assume that $\Pi$ is a non-empty set of *propositional symbols*, $Q$ a non-empty and finite set of *states*, and $\mathbb{A}\mathrm{gt}$ a non-empty set of agents. If not said otherwise we assume that

$$\mathbb{A}\mathrm{gt} = \{1, \dots, k\}$$

and sometimes, in order to make the examples easier to read, we may also use symbolic names $(a, b, c, \dots)$ when referring to agents. We use $\mathsf{p}$, $\mathsf{r}$, ... (resp. $q$, $q'$, $q_1$, ...) as typical representatives for propositions (resp. states).

*Remark 2.2 (Language, Semantics and Logic).* In the following we proceed as follows. We introduce a logical language, say $\mathcal{L}$, which is defined as a set of formulae. We write $\mathcal{L}(P_1, P_2, \dots)$ to emphasise that the language is built over parameters $P_1$, $P_2, \dots$. However, if the parameters are clear from context we omit them; for example, we use $\mathcal{L}$ as a shorthand for $\mathcal{L}(\Pi)$. Elements of $\mathcal{L}$ are called $\mathcal{L}$-formulae. Then, we consider (possibly several) semantics for the language. We refer to each tuple consisting of a language and a suitable semantics over a class of models as a *logic*. We can thus consider the logic as the set of valid formulae over the specified semantics and class of models.

The logic **CTL**, for instance, is given by the language $\mathcal{L}_{CTL}$ using the standard Kripke semantics.

## 2.1 Linear and Branching Time Logics

We begin by recalling two well-known classes of temporal logics: The *linear time* logic **LTL** and the *branching time logics* **CTL** and **CTL**$^\star$.

### 2.1.1 The Languages $\mathcal{L}_{LTL}$, $\mathcal{L}_{CTL}$, and $\mathcal{L}_{CTL^*}$

$\mathcal{L}_{LTL}$ [Pnueli, 1977] extends the language of propositional logic with operators that allow to express temporal patterns over an infinite sequences of states, called *paths*. The basic temporal operators are $\mathcal{U}$ (*until*) and $\bigcirc$ (*in the next state*).

**Definition 2.3 (Language $\mathcal{L}_{LTL}$ [Pnueli, 1977]).** *The* language $\mathcal{L}_{LTL}(\Pi)$ *is given by all formulae generated by the following grammar, where* $\mathsf{p} \in \Pi$ *is a proposition:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mathcal{U} \varphi \mid \bigcirc\varphi.$$

$\mathcal{L}_{LTL}$-formula $\bigcirc(\varphi \wedge \psi)$, for instance, expresses that $\varphi$ and $\psi$ hold in the next moment; $\varphi \mathcal{U} \psi$ states that property $\varphi$ is true at least until $\psi$ becomes true which will eventually be the case. The additional operators $\Diamond$ (*sometime from now on*) and $\Box$ (*always from now on*) can be defined as macros by $\Diamond\varphi \equiv \top\mathcal{U}\varphi$ and $\Box\varphi \equiv \neg\Diamond\neg\varphi$, respectively. The standard Boolean connectives $\top, \bot, \vee, \rightarrow$, and $\leftrightarrow$ are defined in their usual way.

The logic is called *linear time* since formulae are interpreted over infinite *linear* orders of states. CTLs [Emerson and Halpern, 1986] explicitly refer to patterns of properties that can occur along a particular temporal path, *as well as* to the set of possible time series, and thus extend **LTL** by new branching time operators. The latter dimension is handled by *path quantifiers*: $\mathsf{E}$ (*there is a path*) and $\mathsf{A}$ (*for all paths*) where the quantifier $\mathsf{A}$ is defined as macro: $\mathsf{A}\varphi \equiv \neg\mathsf{E}\neg\varphi$. The language of **CTL**$^\star$, $\mathcal{L}_{CTL^*}$, extends $\mathcal{L}_{LTL}$ by adding the existential path quantifier $\mathsf{E}$.

**Definition 2.4 (Language $\mathcal{L}_{CTL^*}$ [Emerson and Halpern, 1986]).** *The* language $\mathcal{L}_{CTL^*}(\Pi)$ *is given by all formulae generated by the following grammar:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{E}\gamma \textit{ where } \gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \gamma\mathcal{U}\gamma \mid \bigcirc\gamma$$

*and* $\mathsf{p} \in \Pi$. *Formulae* $\varphi$ *(resp.* $\gamma$*) are called* state *(resp.* path*) formulae.*

Additionally, the same abbreviations as for $\mathcal{L}_{LTL}$ are defined. $\mathcal{L}_{CTL^*}$-formula $\mathsf{E}\Diamond\varphi$, for instance, ensures that there is at least one path on which $\varphi$ holds now or at some future time moment. Thus, $\mathcal{L}_{CTL^*}$-formulae do not only talk about temporal patterns on a given path but also quantify (existentially or universally) over such paths.

Finally, we define a fragment of **CTL**$^\star$ called **CTL** [Clarke and Emerson, 1981] which is strictly *less expressive* but has *better computational properties.* The language $\mathcal{L}_{CTL}$ restricts $\mathcal{L}_{CTL^*}$ in such a way that each temporal operator must be directly preceded by a path quantifier. For example, $\mathsf{A}\Box\mathsf{E}\bigcirc\mathsf{p}$ is an $\mathcal{L}_{CTL}$-formula whereas $\mathsf{A}\Box\Diamond\mathsf{p}$ is not. Although this completely characterises the language we also provide the original definition in which modalities are given by path quantifiers *coupled* with temporal operators. Note that, chronologically, **CTL** was proposed and studied before **CTL**$^\star$.

**Definition 2.5 (Language $\mathcal{L}_{CTL}$ [Clarke and Emerson, 1981]).** *The* language $\mathcal{L}_{CTL}(\Pi)$ *is given by all formulae generated by the following grammar, where* $\mathsf{p} \in \Pi$*:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{E}(\varphi\mathcal{U}\varphi) \mid \mathsf{E}\bigcirc\varphi \mid \mathsf{E}\Box\varphi.$$

Again, the Boolean connectives are given by their usual abbreviations. In addition to that, we define the following: $\Diamond\varphi \equiv \top\mathcal{U}\varphi$, $\mathsf{A}\bigcirc\varphi \equiv \neg\mathsf{E}\bigcirc\neg\varphi$,

$\mathsf{A}\Box\varphi \equiv \neg\mathsf{E}\Diamond\neg\varphi$, and $\mathsf{A}\varphi\mathcal{U}\psi \equiv \neg\mathsf{E}((\neg\psi)\mathcal{U}(\neg\varphi \wedge \neg\psi)) \wedge \neg\mathsf{E}\Box\neg\psi$. We note that in the definition of the language the existential quantifier cannot be replaced by the universal one without losing expressiveness (cf. [Laroussinie, 1995]).

### 2.1.2 Semantics: LTL, CTL*, and CTL

As mentioned above, the semantics of **LTL** is given over *paths* that are infinite sequences of states from $Q$ and a labelling function $\pi : \Pi \to \mathcal{P}(Q)$ that determines which propositions are true at which states.

**Definition 2.6 (Path $\lambda$).** *A* path $\lambda$ *over a set of states $Q$ is an infinite sequence from $Q^\omega$. We also identify it with a mapping $\mathbb{N}_0 \to Q$. We use $\lambda[i]$ to denote the $i$th position on path $\lambda$ (starting from $i = 0$) and $\lambda[i,\infty] : \mathbb{N}_0 \to Q$ to denote the subpath $\lambda[i,\infty] = \lambda[i]\lambda[i+1]\ldots$ of $\lambda$ starting from $i$, i.e. $\lambda[i,\infty][j] = \lambda[i+j]$ for all $j \in \mathbb{N}_0$.*

**Definition 2.7 (Semantics $\models^{\mathsf{LTL}}$).** *Let $\lambda$ be a path and $\pi : \Pi \to \mathcal{P}(Q)$ be a labelling function. The semantics of $\mathcal{L}_{LTL}$-formulae is defined by the satisfaction relation $\models^{\mathsf{LTL}}$ defined as follows:*

$\lambda, \pi \models^{\mathsf{LTL}} \mathsf{p}$ *iff* $\lambda[0] \in \pi(\mathsf{p})$ *and* $\mathsf{p} \in \Pi$;
$\lambda, \pi \models^{\mathsf{LTL}} \neg\varphi$ *iff not* $\lambda, \pi \models^{\mathsf{LTL}} \varphi$ *(we also write* $\lambda, \pi \not\models^{\mathsf{LTL}} \varphi$*);*
$\lambda, \pi \models^{\mathsf{LTL}} \varphi \wedge \psi$ *iff* $\lambda, \pi \models^{\mathsf{LTL}} \varphi$ *and* $\lambda, \pi \models^{\mathsf{LTL}} \psi$;
$\lambda, \pi \models^{\mathsf{LTL}} \bigcirc\varphi$ *iff* $\lambda[1,\infty], \pi \models^{\mathsf{LTL}} \varphi$*; and*
$\lambda, \pi \models^{\mathsf{LTL}} \varphi\mathcal{U}\psi$ *iff there is an* $i \in \mathbb{N}_0$ *such that* $\lambda[i,\infty], \pi \models \psi$ *and* $\lambda[j,\infty], \pi \models^{\mathsf{LTL}} \varphi$ *for all* $0 \leq j < i$.

Thus, according to Remark 2.2, the *logic* **LTL** is given by $(\mathcal{L}_{LTL}, \models^{\mathsf{LTL}})$. Paths are considered as (canonical) models for $\mathcal{L}_{LTL}$-formulae.

For model checking[1] we require a finite representation of the input $\lambda$. To this end, we use a (pointed) Kripke model $\mathfrak{M}, q$ and consider the problem whether an $\mathcal{L}_{LTL}$-formula holds on *all* paths of $\mathfrak{M}$ starting in $q$.

**Definition 2.8 (Kripke model).** *A* Kripke model *(or* unlabelled transition system*) is given by $\mathfrak{M} = \langle Q, \mathcal{R}, \Pi, \pi \rangle$ where $Q$ is a nonempty set of states (or possible worlds), $\mathcal{R} \subseteq Q \times Q$ is a* serial *transition relation on states, $\Pi$ is a set of atomic propositions, and $\pi : \Pi \to \mathcal{P}(Q)$ is a valuation of propositions. We use $X_{\mathfrak{M}}$ to refer to an element $X$ of $\mathfrak{M}$. Often, we omit the subscript "$\mathfrak{M}$".*

An $\mathfrak{M}$-*path* $\lambda$ (or *computation*) is an infinite sequence of states that refers to a possible course of action.

---

[1] Model checking is the process of checking whether a given formula holds in a given model.

**Definition 2.9 ($\mathfrak{M}$-path, $\Lambda_{\mathfrak{M}}(q)$).** *An* $\mathfrak{M}$-path *(or* $\mathfrak{M}$-computation*) is given by $\lambda \in Q_{\mathfrak{M}}^{\omega}$ such that subsequent states are connected by transitions from $\mathcal{R}_{\mathfrak{M}}$. We use the same notation for these paths as introduced in Definition 2.6. For $q \in Q$ we use $\Lambda_{\mathfrak{M}}(q)$ to denote the set of all $\mathfrak{M}$-paths starting in $q$ and we define $\Lambda_{\mathfrak{M}}$ as $\bigcup_{q \in Q} \Lambda_{\mathfrak{M}}(q)$. We refer to a path from $\Lambda_{\mathfrak{M}}(q)$ as a path $(\mathfrak{M}, q)$-path. The subscript "$\mathfrak{M}$" is often omitted and we refer to an $\mathfrak{M}$-path simply as path when clear from context.*

$\mathcal{L}_{CTL^*}$- and $\mathcal{L}_{CTL}$-formulae are interpreted over Kripke models. In addition to $\mathcal{L}_{LTL}$-(path) formulae (which can only occur as subformulae) it must be specified how state formulae are evaluated.

**Definition 2.10 (Semantics $\models^{\mathsf{CTL}^\star}$).** *Let $\mathfrak{M}$ be a Kripke model, $q \in Q$ and $\lambda \in \Lambda$. The semantics of $\mathcal{L}_{CTL^*}$- and $\mathcal{L}_{CTL}$-formulae are given by the satisfaction relation $\models^{\mathsf{CTL}^\star}$ as follows:*

$\mathfrak{M}, q \models^{\mathsf{CTL}^\star} \mathsf{p}$ *iff* $\lambda[0] \in \pi(\mathsf{p})$ *and* $\mathsf{p} \in \Pi$*;*
$\mathfrak{M}, q \models^{\mathsf{CTL}^\star} \neg\varphi$ *iff* $\mathfrak{M}, q \not\models^{\mathsf{CTL}^\star} \varphi$*;*
$\mathfrak{M}, q \models^{\mathsf{CTL}^\star} \varphi \wedge \psi$ *iff* $\mathfrak{M}, q \models^{\mathsf{CTL}^\star} \varphi$ *and* $\mathfrak{M}, q \models^{\mathsf{CTL}^\star} \psi$*;*
$\mathfrak{M}, q \models^{\mathsf{CTL}^\star} \mathsf{E}\varphi$ *iff there is a path* $\lambda \in \Lambda(q)$ *such that* $\mathfrak{M}, \lambda \models^{\mathsf{CTL}^\star} \varphi$*;*

*and for path formulae by:*

$\mathfrak{M}, \lambda \models^{\mathsf{CTL}^\star} \varphi$ *iff* $\mathfrak{M}, \lambda[0] \models^{\mathsf{CTL}^\star} \varphi$ *for a state formula* $\varphi$*;*
$\mathfrak{M}, \lambda \models^{\mathsf{CTL}^\star} \neg\gamma$ *iff* $\mathfrak{M}, \lambda \not\models^{\mathsf{CTL}^\star} \gamma$*;*
$\mathfrak{M}, \lambda \models^{\mathsf{CTL}^\star} \gamma \wedge \delta$ *iff* $\mathfrak{M}, \lambda \models^{\mathsf{CTL}^\star} \gamma$ *and* $\mathfrak{M}, \lambda \models^{\mathsf{CTL}^\star} \delta$*;*
$\mathfrak{M}, \lambda \models^{\mathsf{CTL}^\star} \bigcirc\gamma$ *iff* $\lambda[1, \infty], \pi \models^{\mathsf{CTL}^\star} \gamma$*; and*
$\mathfrak{M}, \lambda \models^{\mathsf{CTL}^\star} \gamma\mathcal{U}\delta$ *iff there is an* $i \in \mathbb{N}_0$ *such that* $\mathfrak{M}, \lambda[i, \infty] \models^{\mathsf{CTL}^\star} \delta$ *and* $\mathfrak{M}, \lambda[j, \infty] \models^{\mathsf{CTL}^\star} \gamma$ *for all* $0 \le j < i$.

Alternatively, an equivalent *state-based* semantics for **CTL** can be given:

$\mathfrak{M}, q \models^{\mathsf{CTL}} \mathsf{p}$ iff $q \in \pi(\mathsf{p})$ and $\mathsf{p} \in \Pi$;
$\mathfrak{M}, q \models^{\mathsf{CTL}} \neg\varphi$ iff $\mathfrak{M}, q \not\models^{\mathsf{CTL}} \varphi$;
$\mathfrak{M}, q \models^{\mathsf{CTL}} \varphi \wedge \psi$ iff $\mathfrak{M}, q \models^{\mathsf{CTL}} \varphi$ and $\mathfrak{M}, q \models^{\mathsf{CTL}} \psi$;
$\mathfrak{M}, q \models^{\mathsf{CTL}} \mathsf{E} \bigcirc \varphi$ iff there is a path $\lambda \in \Lambda(q)$ such that $\mathfrak{M}, \lambda[1] \models^{\mathsf{CTL}} \varphi$;
$\mathfrak{M}, q \models^{\mathsf{CTL}} \mathsf{E}\Box\varphi$ iff there is a path $\lambda \in \Lambda(q)$ such that $\mathfrak{M}, \lambda[i] \models^{\mathsf{CTL}} \varphi$ for each $i \in \mathbb{N}_0$;
$\mathfrak{M}, q \models^{\mathsf{CTL}} \mathsf{E}\varphi\mathcal{U}\psi$ iff there is a path $\lambda \in \Lambda(q)$ such that $\mathfrak{M}, \lambda[i] \models^{\mathsf{CTL}} \psi$ for some $i \in \mathbb{N}_0$, and $\mathfrak{M}, \lambda[j, \infty] \models^{\mathsf{CTL}} \varphi$ for all $0 \le j < i$.

This equivalent semantics underlies the model checking algorithm for **CTL** which can be implemented in **P** rather than **PSPACE** which is the case for **CTL**$^\star$ (cf. Section 5.2). The logics **CTL** and **CTL**$^\star$ are given by $(\mathcal{L}_{CTL}, \models^{\mathsf{CTL}})$ and $(\mathcal{L}_{CTL^*}, \models^{\mathsf{CTL}^\star})$, respectively.

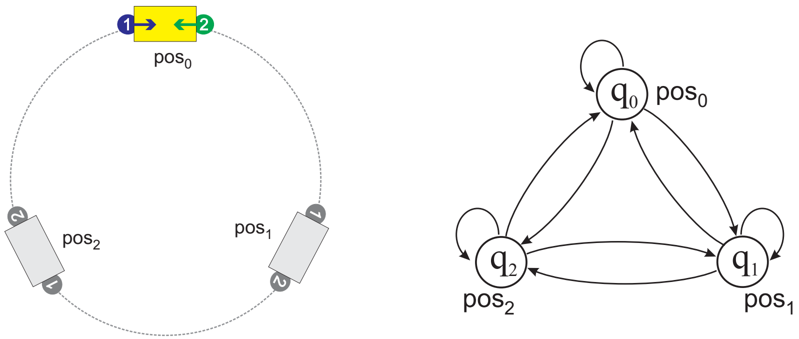



**Fig. 2.1.** Two robots and a carriage: A schematic view (left) and a Kripke model $\mathfrak{M}_0$ that models the scenario (right).

*Remark 2.11.* Note that model checking problem for an $\mathcal{L}_{LTL}$-formula $\varphi$ with respect to a given Kripke model $\mathfrak{M}$ and a state $q$ is equivalent to the $\mathsf{CTL}^\star$ model checking problem $\mathfrak{M}, q \models^{\mathsf{CTL}^\star} \mathsf{A}\varphi$.

We end this section with an example.

*Example 2.12 (Robots and Carriage).* We consider the scenario depicted in Figure 2.1. Two robots push a carriage from opposite sides. As a result, the carriage can move clockwise or anticlockwise, or it can remain in the same place – depending on who pushes with more force (and, perhaps, who refrains from pushing). To make our model of the domain discrete, we identify 3 different positions of the carriage, and associate them with states $q_0$, $q_1$, and $q_2$. The arrows in transition system $\mathfrak{M}_0$ indicate how the state of the system can change in a single step. We label the states with propositions $\mathsf{pos}_0, \mathsf{pos}_1, \mathsf{pos}_2$, respectively, to allow for referring to the current position of the carriage in the object language.

For example, we have $\mathfrak{M}_0, q_0 \models^{\mathsf{CTL}} \mathsf{E}\Diamond\mathsf{pos}_1$: In state $q_0$, there is a path such that the carriage will reach position 1 sometime in the future. Of course, the same is not true for *all* paths, so we also have that $\mathfrak{M}_0, q_0 \models^{\mathsf{CTL}} \neg\mathsf{A}\Diamond\mathsf{pos}_1$.

### 2.1.3 $\mathsf{CTL}^+$

The language $\mathcal{L}_{CTL^+}$ is the subset of $\mathcal{L}_{CTL^*}$ that requires each temporal operator to be followed by a state formula, but path quantifiers are allowed to be followed by a Boolean combinations of path subformulae.

**Definition 2.13 (Language $\mathcal{L}_{CTL^+}$).** *The* language $\mathcal{L}_{CTL^+}(\Pi)$ *is given by all formulae generated by the following grammar:* $\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \land \varphi \mid \mathsf{E}\gamma$ *where* $\gamma ::= \neg\gamma \mid \gamma \land \gamma \mid \varphi\mathcal{U}\varphi \mid \bigcirc\varphi$*, and* $\mathsf{p} \in \Pi$*.*

We define the logic $\mathbf{CTL}^+$ analogously to $\mathbf{CTL}^\star$. We would like to point out that the logic $\mathbf{CTL}^+$ is *not* more expressive than $\mathbf{CTL}$ [Emerson and Halpern, 1985] but allows for an exponentially more succinct presentation [Wilke, 1999]. This more compact representation has its price in terms of model checking; the complexity increases from $\mathbf{P}$ to $\mathbf{\Delta_2^P}$ (cf. Section 5.2).

## 2.2 Alternating Time Temporal Logics

In the following we present alternating time temporal logics (ATLs). These logics can be used to model and to reason about strategic abilities of agents. We consider semantics based on perfect and on imperfect information. Here "perfect information" is understood in such a way that agents know the current state of the system: Agents are able to distinguish all states of the system. This is fundamentally different from the imperfect information setting presented in Section 2.2.3 where *different* states possibly provide the same information to an agent and thus make them appear indistinguishable to it. This must be reflected in the agents' available strategies.

### 2.2.1 The Languages $\mathcal{L}_{ATL^*}$ and $\mathcal{L}_{ATL}$

ATLs [Alur et al., 1997, 2002] are generalisations of CTLs. In $\mathcal{L}_{ATL^*}$/ $\mathcal{L}_{ATL}$ the path quantifiers $\mathsf{E}, \mathsf{A}$ are replaced by *cooperation modalities* $\langle\langle A\rangle\rangle$ where $A \subseteq \mathbb{A}\mathrm{gt}$ is a team of agents. Formula $\langle\langle A\rangle\rangle\gamma$ expresses that team $A$ has a *collective strategy* to enforce $\gamma$. The definition of the language is given below.

**Definition 2.14 (Language $\mathcal{L}_{ATL^*}$ [Alur et al., 1997]).** *The* language $\mathcal{L}_{ATL^*}(\Pi, \mathbb{A}\mathrm{gt})$ *is given by all formulae generated by the following grammar:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \land \varphi \mid \langle\langle A\rangle\rangle\gamma \text{ where } \gamma ::= \varphi \mid \neg\gamma \mid \gamma \land \gamma \mid \gamma\mathcal{U}\gamma \mid \bigcirc\gamma,$$

$A \subseteq \mathbb{A}\mathrm{gt}$*, and* $\mathsf{p} \in \Pi$*. Formulae* $\varphi$ *(resp.* $\gamma$*) are called* state *(resp.* path*) formulae.*

We use similar abbreviations to the ones introduced in Section 2.1.1. In the case of a single agent $a$ we will also write $\langle\langle a\rangle\rangle$ instead of $\langle\langle\{a\}\rangle\rangle$. An example of an $\mathcal{L}_{ATL^*}$-formula is $\langle\langle A\rangle\rangle\Box\Diamond\mathsf{p}$ which expresses that coalition $A$ can guarantee that $\mathsf{p}$ is satisfied infinitely many times (ever and ever again in the future).

The language $\mathcal{L}_{ATL}$ restricts $\mathcal{L}_{ATL^*}$ in the same way as $\mathcal{L}_{CTL}$ restricts $\mathcal{L}_{CTL^*}$. Each temporal operator must be directly preceded by a cooperation modality.

**Definition 2.15 (Language $\mathcal{L}_{ATL}$ [Alur et al., 1997]).** *The* language $\mathcal{L}_{ATL}(\Pi, \mathbb{A}\mathrm{gt})$ *is given by all formulae generated by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle \bigcirc \varphi \mid \langle\langle A \rangle\rangle \Box \varphi \mid \langle\langle A \rangle\rangle \varphi \mathcal{U} \varphi$$

*where* $A \subseteq \mathbb{A}\mathrm{gt}$ *and* $\mathsf{p} \in \Pi$.

$\mathcal{L}_{ATL^*}$-formula $\langle\langle A \rangle\rangle \Box \Diamond \mathsf{p}$ is obviously not a formula of $\mathcal{L}_{ATL}$ as it includes two consecutive temporal operators. In more general terms, $\mathcal{L}_{ATL}$ does not allow to express abilities related to, e.g., fairness properties. Still, many interesting properties are expressible. For instance, we can state that agent $a$ has a strategy that permanently takes away the ability to enforce $\bigcirc \mathsf{p}$ from coalition $B$: $\langle\langle a \rangle\rangle \Box \neg \langle\langle B \rangle\rangle \bigcirc \mathsf{p}$. As for the two computation tree logics, the choice between $\mathcal{L}_{ATL^*}$ and $\mathcal{L}_{ATL}$ reflects a tradeoff between expressiveness and practicality.

### 2.2.2 Perfect Information Semantics: $\mathsf{ATL}_{\mathsf{Iy}}$, $\mathsf{ATL}^*_{\mathsf{Iy}}$

The semantics for $\mathcal{L}_{ATL^*}$ and $\mathcal{L}_{ATL}$ are defined over a variant of transition systems where transitions are labeled with combinations of actions, one per agent.

**Definition 2.16** (CGS)**.** *A* concurrent game structure *(*CGS*) is a tuple*

$$\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o \rangle$$

*which includes a nonempty finite set of all agents* $\mathbb{A}\mathrm{gt} = \{1, \ldots, k\}$*, a nonempty set of states* $Q$*, a set of atomic propositions* $\Pi$ *and their valuation* $\pi : \Pi \to \mathcal{P}(Q)$*, and a nonempty finite set of (atomic) actions* $Act$*. Function* $d : \mathbb{A}\mathrm{gt} \times Q \to \mathcal{P}(Act)$ *defines nonempty sets of actions available to agents at each state, and* $o$ *is a (deterministic) transition function that assigns the outcome state* $q' = o(q, \alpha_1, \ldots, \alpha_k)$ *to state* $q$ *and a tuple of actions* $\langle \alpha_1, \ldots, \alpha_k \rangle$ *for* $\alpha_i \in d(i, q)$ *and* $1 \le i \le k$*, that can be executed by* $\mathbb{A}\mathrm{gt}$ *in* $q$*. We also write* $d_a(q)$ *instead of* $d(a, q)$*.*

It is assumed that all the agents execute their actions synchronously: The combination of actions together with the current state determines the next transition of the system.

A *strategy* of agent $a$ is a conditional plan that specifies what $a$ is going to do in each situation. It makes sense, from a conceptual and computational point of view, to distinguish between two types of “situations” (and hence strategies): An agent might base its decision only on the current state or on the whole history of events that have happened. A *history* is considered as a finite sequence of states of the system.

**Definition 2.17 (*IR*- and *Ir*-strategies).** *A* perfect information perfect recall strategy *for agent $a$ (IR*-strategy *for short)*[2] *is a tuple $(s_a, a)$ where $s_a : Q^+ \to Act$ is a function such that $s_a(q_0 q_1 \dots q_n) \in d_a(q_n)$. The set of such strategies is denoted by $\Sigma_a^{IR}$.*

*A* perfect information memoryless strategy *for agent $a$ (Ir*-strategy *for short) is given by a tuple $(s_a, a)$ consisting of a function $s_a : Q \to Act$ where $s_a(q) \in d_a(q)$. The set of such strategies is denoted by $\Sigma_a^{Ir}$. We will use the term* strategy *to refer to any of these two types.*

*Remark 2.18 (Notation for strategies).* In the following we shall identify $(s_a, a)$ with $s_a$. However, formally one has to assume that strategies are given as tuples. For, otherwise one would not be able to "select" an agent's strategy from a set of strategies (cf. Definition 2.19).

We will also consider a memoryless strategy as a perfect recall strategy satisfying $s(hq) = s(h'q)$ for all $h, h' \in Q^*$.

**Definition 2.19 (Collective strategy, $s_A|_a$, $s_A|_B$, $s_\emptyset$).** *A* collective strategy *(memoryless or perfect recall) for a group of agents $A = \{a_1, \dots, a_r\} \subseteq \mathbb{A}\mathrm{gt}$ is a set*

$$s_A = \{s_a \mid a \in A, s_a \text{ is a strategy for } a\}$$

*of strategies, one per agent from $A$. The set of $A$'s collective perfect information strategies is given by $\Sigma_A^{IR} = \prod_{a \in A} \Sigma_a^{IR}$ (in the perfect recall case) and $\Sigma_A^{Ir} = \prod_{a \in A} \Sigma_a^{Ir}$ (in the memoryless case). The set of all* (complete) strategy profiles *is given by $\Sigma^{IR} = \Sigma_{\mathbb{A}\mathrm{gt}}^{IR}$ (resp. $\Sigma^{Ir} = \Sigma_{\mathbb{A}\mathrm{gt}}^{Ir}$).*

*By $s_A|_a$, we denote agent $a$'s strategy $s_a$ of the collective strategy $s_A$ where $a \in A$; i.e. $s_A|_a \in s_A \cap \Sigma_a^{Ix}$ for $x \in \{R, r\}$. For a group $B$ of agents we use $s_A|_B$ to refer to $B$'s collective substrategy; i.e.*

$$s_A|_B := s_A \cap \Sigma_B.$$

*We use the special strategy $s_\emptyset$ to refer to the empty strategy $\emptyset$.*

*Remark 2.20 (Strategies).*

(a) For convenience we sometimes refer to a collective strategy as a tuple.
(b) We note that there is a formal difference between $s_a$ and $s_{\{a\}}$. The former is a strategy the latter is a set containing a strategy.

Function $out(q, s_A)$ returns the set of all paths that may occur when agents $A$ execute strategy $s_A$ from state $q$ onward.

[2] The notation was introduced in [Schobbens, 2004] where $i$ (resp. $I$) stands for *imperfect* (resp. *perfect*) *information* and $r$ (resp. $R$) for *imperfect* (resp. *perfect*) *recall.* Also compare with Section 2.2.3.

**Definition 2.21 (Outcome).** *The* outcome $out_{\mathfrak{M}}(q, s_A)$ of $s_A$ from state $q$ in model $\mathfrak{M}$ *is the set of all paths* $\lambda = q_0q_1q_2\ldots$ *such that* $q_0 = q$ *and for each* $i = 1, 2, \ldots$ *there exists a tuple of agents' decisions* $\langle \alpha_{a_1}^{i-1}, \ldots, \alpha_{a_k}^{i-1}\rangle$ *such that* $\alpha_a^{i-1} \in d_a(q_{i-1})$ *for every* $a \in \mathbb{A}\text{gt}$, *and* $\alpha_a^{i-1} = s_A|_a(q_0q_1\ldots q_{i-1})$ *for every* $a \in A$, *and* $o(q_{i-1}, \alpha_{a_1}^{i-1}, \ldots, \alpha_{a_k}^{i-1}) = q_i$.

*In line with Remark 2.20 we identify* $out_{\mathfrak{M}}(q, s_a)$ *with* $out_{\mathfrak{M}}(q, \{s_a\})$. *Often, we will omit the subscript "*$\mathfrak{M}$*" if clear from context. For an Ir-strategy* $s_A$ *the outcome is defined analogously: "*$s_A|_a(q_0q_1\ldots q_{i-1})$*" is simply replaced by "*$s_A|_a(q_{i-1})$*".*

The semantics for $\mathcal{L}_{ATL}$ and $\mathcal{L}_{ATL^*}$, one for each type of strategy, are shown below. Informally speaking, $\mathfrak{M}, q \models \langle\langle A\rangle\rangle\gamma$ if, and only if, there exists a collective strategy $s_A$ such that $\gamma$ holds for all computations from $out(q, s_A)$.

The semantics is defined in a similar way to $\models^{\mathsf{CTL}^*}$ from Definition 2.10 only the rule for $\mathsf{E}\varphi$ is replaced.

**Definition 2.22 (Perfect information semantics $\models_{IR}$ and $\models_{Ir}$).** *Let* $\mathfrak{M}$ *be a* CGS. *The* perfect information perfect recall semantics *for* $\mathcal{L}_{ATL^*}$ *and* $\mathcal{L}_{ATL}$, *IR*-semantics *for short, denoted by* $\models_{IR}$, *is defined by the following clauses:*

$\mathfrak{M}, q \models_{IR} \mathsf{p}$ *iff* $\lambda[0] \in \pi(\mathsf{p})$ *and* $\mathsf{p} \in \Pi$;
$\mathfrak{M}, q \models_{IR} \neg\varphi$ *iff* $\mathfrak{M}, q \not\models_{IR} \varphi$;
$\mathfrak{M}, q \models_{IR} \varphi \wedge \psi$ *iff* $\mathfrak{M}, q \models_{IR} \varphi$ *and* $\mathfrak{M}, q \models_{IR} \psi$;
$\mathfrak{M}, q \models_{IR} \langle\langle A\rangle\rangle\gamma$ *iff there is an IR-strategy* $s_A \in \Sigma_A^{IR}$ *for* $A$ *such that for every path* $\lambda \in out(q, s_A)$, *we have* $\mathfrak{M}, \lambda \models_{IR} \gamma$;

*and for path formulae by:*

$\mathfrak{M}, \lambda \models_{IR} \varphi$ *iff* $\mathfrak{M}, \lambda[0] \models_{IR} \varphi$;
$\mathfrak{M}, \lambda \models_{IR} \neg\gamma$ *iff* $\mathfrak{M}, \lambda \not\models_{IR} \gamma$;
$\mathfrak{M}, \lambda \models_{IR} \gamma \wedge \delta$ *iff* $\mathfrak{M}, \lambda \models_{IR} \gamma$ *and* $\mathfrak{M}, \lambda \models_{IR} \delta$;
$\mathfrak{M}, \lambda \models_{IR} \bigcirc\gamma$ *iff* $\lambda[1, \infty], \pi \models_{IR} \gamma$; *and*
$\mathfrak{M}, \lambda \models_{IR} \gamma\mathcal{U}\delta$ *iff there is an* $i \in \mathbb{N}_0$ *such that* $\mathfrak{M}, \lambda[i, \infty] \models_{IR} \delta$ *and* $\mathfrak{M}, \lambda[j, \infty] \models_{IR} \gamma$ *for all* $0 \leq j < i$.

*The* perfect information memoryless semantics *for* $\mathcal{L}_{ATL^*}$ *and* $\mathcal{L}_{ATL}$, Ir-semantics *for short, is given as above but "IR" is replaced by "Ir" everywhere.*

*Remark 2.23.* We note that cooperation modalities are neither "diamonds" nor "boxes" in terms of classical modal logic. Rather, they are combinations of both as their structure can be described by "$\exists\forall$": We ask for the *existence* of a strategy of the proponents which is successful against *all* responses of the opponents.

In [Broersen et al., 2006] it was shown how the cooperation modalities can be decomposed into two parts in the context of **STIT** logic. A similar decomposition is considered in [Jamroga, 2008b] for the analysis of stochastic multi-agent systems.

The $\mathcal{L}_{CTL^*}$ path quantifiers $\mathsf{A}$ and $\mathsf{E}$ can be embedded in $\mathcal{L}_{ATL^*}$ using the *IR*-semantics in the following way: $\mathsf{A}\gamma \equiv \langle\langle\varnothing\rangle\rangle\gamma$ and $\mathsf{E}\gamma \equiv \langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\gamma$.

Analogously to **CTL**, it is possible to provide a state-based semantics for $\mathcal{L}_{ATL}$. We only present the clause for $\langle\langle A\rangle\rangle\Box\varphi$ (the cases for the other temporal operators are given in a similar way):

$\mathfrak{M}, q, \models^{\mathsf{ATL}}_{Ix} \langle\langle A\rangle\rangle\Box\varphi$ iff there is an $Ix$-strategy $s_A \in \Sigma^{Ix}_A$ such that for all $\lambda \in out(q, s_A)$ and $i \in \mathbb{N}_0$ it holds that $\mathfrak{M}, q, \models^{\mathsf{ATL}}_{Ix} \varphi$

where $x$ is either $R$ or $r$.

This already suggests that dealing with $\mathcal{L}_{ATL}$ is computationally less expensive than with $\mathcal{L}_{ATL^*}$. On the other hand, $\mathcal{L}_{ATL}$ lacks expressiveness: There is no formula which is true for the memoryless semantics and false for the perfect recall semantics, and vice versa.

**Theorem 2.24.** [3] *For $\mathcal{L}_{ATL}$, the perfect recall semantics is equivalent to the memoryless semantics under perfect information, i.e., $\mathfrak{M}, q \models_{IR} \varphi$ iff $\mathfrak{M}, q \models_{Ir} \varphi$. Both semantics are different for $\mathcal{L}_{ATL^*}$.*

Thus, when referring to $\mathcal{L}_{ATL}$ using the perfect information semantics, we can omit the subscript in the satisfaction relation $\models$.

**Definition 2.25 ($\mathsf{ATL_{Ix}}$, $\mathsf{ATL^*_{Ix}}$, ATL, $\mathsf{ATL^*}$).** *We define $\mathsf{ATL_{Ix}}$ and $\mathsf{ATL^*_{Ix}}$ as the logics $(\mathcal{L}_{ATL}, \models_{Ix})$ and $(\mathcal{L}_{ATL^*}, \models_{Ix})$ where $x \in \{r, R\}$, respectively. Moreover, we use* **ATL** *(resp. $\mathsf{ATL^*}$) as abbreviation for $\mathsf{ATL_{IR}}$ (resp. $\mathsf{ATL^*_{IR}}$).*

We note that $\mathsf{ATL_{IR}}$ and $\mathsf{ATL_{Ir}}$ are equivalent logics (i.e. their sets of validities coincide). We end our presentation of the language and semantics with an example.

*Example 2.26 (Robots and Carriage, ctd.).* Transition system $\mathfrak{M}_0$ from Figure 2.1 enabled us to study the evolution of the system as a whole. However, it did not allow us to represent *who* can achieve *what*, and how the possible actions of the agents interact. Concurrent game structure $\mathfrak{M}_1$, presented in Figure 2.2, fills the gap. We assume that each robot can either push (action *push*) or refrain from pushing (action *wait*). Moreover, they both use the same force when pushing. Thus, if the robots push simultaneously or wait simultaneously, the carriage does not move. When only one of the robots is pushing, the carriage moves accordingly.

[3] The property has been first observed in [Schobbens, 2004] but it follows from [Alur et al., 2002] in a straightforward way.

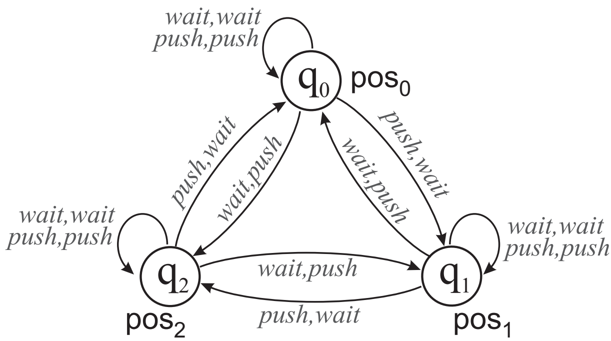



**Fig. 2.2.** The robots and the carriage: A concurrent game structure $\mathfrak{M}_1$.

As the outcome of each robot's action depends on the current action of the other robot, no agent can make sure that the carriage moves to any particular position. So, we have for example that $\mathfrak{M}_1, q_0 \models \neg\langle\langle 1\rangle\rangle\Diamond \mathsf{pos}_1$. On the other hand, the agent can at least make sure that the carriage will *avoid* particular positions. For instance, it holds that $\mathfrak{M}_1, q_0 \models \langle\langle 1\rangle\rangle\Box\neg\mathsf{pos}_1$, the right strategy being $s_1(q_0) = wait, s_1(q_2) = push$ (the action that we specify for $q_1$ is irrelevant).

### 2.2.3 Imperfect Information Semantics: $\mathbf{ATL_{iy}}$, $\mathbf{ATL^*_{iy}}$

The logics introduced so far include no way of addressing uncertainty that an agent or a process may have about the current situation. Several extensions capable of dealing with imperfect information have been proposed, e.g., in [Alur et al., 2002; Schobbens, 2004; Jamroga and Ågotnes, 2007].

Here, we take Schobbens' version from [Schobbens, 2004] as the "core", minimal $\mathcal{L}_{ATL^*}$-based language for strategic ability under imperfect information. We use the already defined languages $\mathcal{L}_{ATL^*}$ and $\mathcal{L}_{ATL}$ but here the cooperation modalities have an additional *epistemic flavour* by means of a modified semantics as we shall show below.[4] The models can be seen as CGSs augmented with a family of indistinguishability relations $\sim_a \subseteq Q \times Q$, one per agent $a \in \mathbb{A}\mathrm{gt}$. The relations describe agents' uncertainty: $q \sim_a q'$ means that agent $a$ cannot distinguish between states $q$ and $q'$ of the system. Each $\sim_a$ is assumed to be an equivalence relation. It is also required that agents have the same choices in indistinguishable states.

[4] In [Schobbens, 2004] the cooperation modalities are presented with a subscript: $\langle\langle A\rangle\rangle_{ir}$ to indicate that they address agents with imperfect <u>**i**</u>nformation and imperfect <u>**r**</u>ecall. Here, we take on a rigorous semantic point of view and keep the syntax unchanged.

**Definition 2.27 (ICGS).** *An* imperfect information concurrent game structure *(*ICGS*) is given by*

$$\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \{\sim_a \mid a \in \mathbb{A}\mathrm{gt}\}\rangle$$

*where* $\langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o\rangle$ *is a* CGS*, each* $\sim_a \subseteq Q \times Q$ *is an equivalence relation, and if* $q \sim_a q'$ *then* $d(a,q) = d(a,q')$*.*

Indistinguishability between states can now be extended to histories which are finite sequences of states. Such finite sequences of states appear identical to an agent if they have the same length and provide the same information to the agent at each step.

**Definition 2.28 (Indistinguishable histories).** *Two* histories $h = q_0q_1 \dots q_n \in Q^+$ *and* $h' = q'_0q'_1 \dots q'_{n'} \in Q^+$ *are said to be* indistinguishable *for agent* $a$*,* $h \sim_a h'$*, if and only if,* $n = n'$ *and* $q_i \sim_a q'_i$ *for* $i = 0, 1, \dots, n$*.*

This means, according to the definition of indistinguishability we deal with the synchronous notion of recall according to the classification in [Fagin et al., 1995].

An *imperfect information strategy*[5]–memoryless or perfect recall–of agent $a$ is a plan that takes into account $a$'s epistemic limitations. An executable strategy must prescribe the *same choices for indistinguishable situations.* Therefore, we restrict the strategies that can be used by agents in the following way.

**Definition 2.29 (*iR*-, *ir*-strategies).** *An* imperfect information perfect recall strategy *(iR-strategy for short) of agent* $a$ *is an IR-strategy satisfying the following additional constraint: For all histories* $h, h' \in Q^+$*, if* $h \sim_a h'$ *then* $s_a(h) = s_a(h')$*.*

*An* imperfect information memoryless strategy *(ir-strategy for short) is an Ir-strategy satisfying the following constraint: if* $q \sim_a q'$ *then* $s_a(q) = s_a(q')$*.*

*The set of* $a$*'s ir (resp. iR) strategies is denoted by* $\Sigma_a^{ir}$ *(resp.* $\Sigma_a^{iR}$*). A* collective *iR*/*ir*-strategy *is a combination of individual iR/ir-strategies. The set of A's collective imperfect information strategies is given by* $\Sigma_A^{iR} = \prod_{a \in A} \Sigma_a^{iR}$ *(in the perfect recall case) and* $\Sigma_A^{ir} = \prod_{a \in A} \Sigma_a^{ir}$ *(in the memoryless case). The set of all strategy profiles is given by* $\Sigma^{iR} = \Sigma_{\mathbb{A}\mathrm{gt}}^{iR}$ *(resp.* $\Sigma^{ir} = \Sigma_{\mathbb{A}\mathrm{gt}}^{ir}$*).*

*We use the same notation as introduced in Definition 2.19.*

That is, an *iR*-strategy is required to assign the same actions to indistinguishable histories. As before, a perfect recall strategy (memoryless or not) assigns an action to each element from $Q^+$.

The outcome function $out(q, s_A)$ for the imperfect information cases is defined as before (cf. Definition 2.21).

[5] Also called *uniform* strategy.

**Definition 2.30 (Imperfect information semantics $\models_{iR}$ and $\models_{ir}$).** *Let* $\mathfrak{M}$ *be an* ICGS*, and let* $\text{img}(q,\rho) = \{q' \mid \rho(q,q')\}$ *be the image of state* $q$ *wrt. a binary relation* $\rho$*. The* imperfect information perfect recall *semantics (iR-*semantics*) for* $\mathcal{L}_{ATL^*}$ *and* $\mathcal{L}_{ATL}$*, denoted by* $\models_{iR}$*, is given as in Definition 2.22 with the rule for* $\langle\!\langle A\rangle\!\rangle\gamma$ *replaced by the following clause:*

$\mathfrak{M}, q \models_{iR} \langle\!\langle A\rangle\!\rangle\gamma$ *iff there is an iR-strategy* $s_A \in \Sigma_A^{iR}$ *such that, for each* $q' \in \text{img}(q, \sim_A)$ *and each* $\lambda \in out(s_A, q')$*, we have* $\mathfrak{M}, \lambda \models_{iR} \gamma$ *(where* $\sim_A := \bigcup_{a\in A} \sim_a$*).*

*The* imperfect information memoryless semantics *for* $\mathcal{L}_{ATL^*}$ *and* $\mathcal{L}_{ATL}$*, ir-*semantics for short*, is given as above but "iR" is replaced by "ir" everywhere.*

Note that $\mathfrak{M}, q \models_{ix} \langle\!\langle A\rangle\!\rangle\gamma$ requires $A$ to have a *single* strategy that is successful in *all* states indistinguishable from $q$.

*Remark 2.31 (Implicit knowledge operators).* We note that some knowledge operators are *implicitly* given by the cooperation modalities if the imperfect information semantics is used. In this setting a formula $\langle\!\langle A\rangle\!\rangle\gamma$ is read as follows: Each agent in $A$ *knows* that they (the agents in $A$) have a collective strategy to enforce $\gamma$. In particular, one can express $K_a\varphi$ ("$a$ knows that $\varphi$") by $\langle\!\langle a\rangle\!\rangle\varphi\mathcal{U}\varphi$, and $E_A\varphi$ ("everybody in $A$ knows that $\varphi$") by $\langle\!\langle A\rangle\!\rangle\varphi\mathcal{U}\varphi$. More sophisticated epistemic versions of **ATL** which contain explicit knowledge operators (including ones for common and distributed knowledge) are, for instance, considered in [Jamroga and Ågotnes, 2007; van der Hoek and Wooldridge, 2003; van Otterloo et al., 2003; Goranko and Jamroga, 2004]. We present some of these settings in Section 2.3.2.

**Definition 2.32 ($\mathbf{ATL_{ix}}$, $\mathbf{ATL^*_{ix}}$).** *We define* $\mathbf{ATL_{ix}}$ *and* $\mathbf{ATL^*_{ix}}$ *as the logics* $(\mathcal{L}_{ATL}, \models_{ix})$ *and* $(\mathcal{L}_{ATL^*}, \models_{ix})$ *where* $x \in \{r, R\}$*, respectively.*

*Example 2.33 (Robots and Carriage, ctd.).* We refine the scenario from Examples 2.12 and 2.26 by restricting perception of the robots. Namely, we assume that robot 1 is only able to observe the colour of the surface on which it is standing, and robot 2 perceives only the texture (cf. Figure 2.3). As a consequence, the first robot can distinguish between position 0 and position 1, but positions 0 and 2 look the same to it. Likewise, the second robot can distinguish between positions 0 and 2, but not 0 and 1. We also assume that the agents are memoryless, i.e., they cannot memorise their previous observations.

With their observational capabilities restricted in such way, no agent can make the carriage reach or avoid any selected states singlehandedly. E.g., we have that $\mathfrak{M}_2, q_0 \models_{ir} \neg\langle\!\langle 1\rangle\!\rangle\Box\neg\mathsf{pos}_1$. Note in particular that strategy $s_1$ from Example 2.26 cannot be used here because it is not uniform (indeed, the strategy tells robot 1 to wait in $q_0$ and push in $q_2$ but both states look

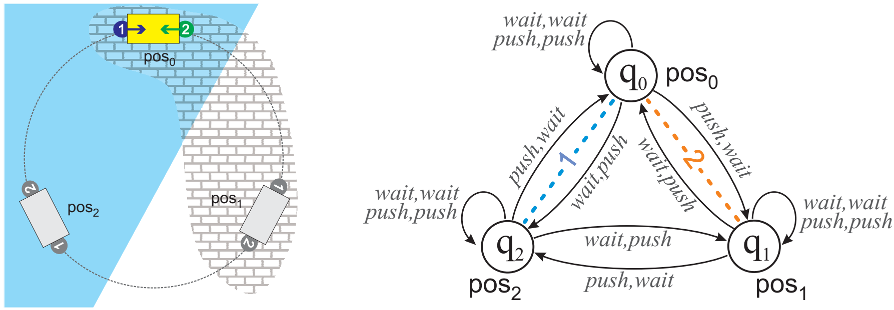



**Fig. 2.3.** Two robots and a carriage: A schematic view (left) and an imperfect information concurrent game structure $\mathfrak{M}_2$ that models the scenario (right).

the same to the robot). The robots cannot even be sure to achieve the task together: $\mathfrak{M}_2, q_0 \models_{ir} \neg\langle\langle 1,2\rangle\rangle\Box\mathsf{pos}_1$ (when in $q_0$, robot 2 considers it possible that the current state of the system is $q_1$, in which case all the hope is gone). So, do the robots know how to play to achieve anything? Yes, for example they know how to make the carriage *reach* a particular state eventually: $\mathfrak{M}_2, q_0 \models_{ir} \langle\langle 1,2\rangle\rangle\Diamond\mathsf{pos}_1$ etc. – it suffices that one of the robots pushes all the time and the other waits all the time. Still, $\mathfrak{M}_2, q_0 \models_{ir} \neg\langle\langle 1,2\rangle\rangle\Diamond\Box\mathsf{pos}_\mathsf{x}$ (for $x = 0,1,2$): there is no memoryless strategy for the robots to bring the carriage to a particular position and keep it there forever.

Most of the above properties hold for the $iR$-semantics as well. Note, however, that for robots with perfect recall we do have that $\mathfrak{M}_2, q_0 \models_{iR} \langle\langle 1,2\rangle\rangle\Diamond\Box\mathsf{pos}_\mathsf{x}$. The right strategy is that one robot pushes and the other waits for the first 3 steps. After that, they know their current position exactly, and can go straight to the specified position.

### 2.2.4 Coalition Logic CL

*Coalition logic* (CL), introduced in [Pauly, 2002], is another logic for modelling and reasoning about strategic abilities of agents. The main construct of CL, $[A]\varphi$, expresses that coalition $A$ can bring about $\varphi$ in a single-step game.

**Definition 2.34 (Language $\mathcal{L}_{CL}$ [Pauly, 2002]).** *The* language $\mathcal{L}_{CL}(\Pi, \mathbb{A}\mathrm{gt})$ *is given by all formulae generated by the following grammar:* $\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid [A]\varphi$, *where* $\mathsf{p} \in \Pi$ *and* $A \subseteq \mathbb{A}\mathrm{gt}$.

In [Pauly, 2002], *coalitional models* have been chosen as semantics for $\mathcal{L}_{CL}$. These models are given by $(Q, E, \pi)$ consisting of a set of states $Q$, a *playable effectivity function* $E$, and a valuation function $\pi$. The effectivity function determines the outcomes that a coalition can guarantee to achieve, i.e., given a set $X \subseteq Q$ of states a coalition $C$ is said to be effective for $X$ iff it can enforce the next state to be in $X$. However, in [Goranko and Jamroga, 2004] it has been shown that CGSs provide an equivalent semantics, and that CL can be seen as the *next-time fragment* of **ATL**. Hence, for this presentation we will interpret $\mathcal{L}_{CL}$-formulae over CGSs, and consider $[A]\varphi$ as an abbreviation for $\langle\langle A\rangle\rangle \bigcirc \varphi$. The various logics $\mathbf{CL_{xy}}$ that we obtain using the semantics $\models_{xy}$ for $x \in \{i, I\}$ and $y \in \{r, R\}$ are defined analogously to $\mathbf{ATL_{xy}}$.

### 2.2.5 ATL$^+$

The language $\mathcal{L}_{ATL^+}$ is the subset of $\mathcal{L}_{ATL^*}$ that requires each temporal operator to be followed by a state formula, but cooperation modalities are allowed to be followed by Boolean combinations of $\mathcal{L}_{ATL}$-based path formulae. Formula $\langle\langle A\rangle\rangle(\Box \mathsf{p} \wedge \Diamond \mathsf{q})$, for instance, is an $\mathcal{L}_{ATL^+}$-formula but not an $\mathcal{L}_{ATL}$-formula. Formally, the language is given as follows:

**Definition 2.35 (Language $\mathcal{L}_{ATL^+}$).** *The* language $\mathcal{L}_{ATL^+}(\Pi, \mathbb{A}\text{gt})$ *is given by all formulae generated by the following grammar:* $\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle\gamma$ *where* $\gamma ::= \neg\gamma \mid \gamma \wedge \gamma \mid \varphi\mathcal{U}\varphi \mid \bigcirc\varphi$, $A \subseteq \mathbb{A}\text{gt}$ *and* $\mathsf{p} \in \Pi$.

We define the various logics emerging from $\mathcal{L}_{ATL^+}$ together with one of the introduced semantics analogously to the case of $\mathcal{L}_{ATL^*}$. The logic $\mathbf{ATL^+_{IR}}$ is strictly more expressive than $\mathbf{ATL_{IR}}$ (contrary to common belief, each $\mathbf{ATL^+_{IR}}$ formula can only be translated to an equivalent $\mathbf{ATL_{IR}}$ formula if the "release" or "weak until" operator is added to the language of $\mathcal{L}_{ATL}$ [Bulling and Jamroga, 2010a; Laroussinie et al., 2008; Harding et al., 2002]) but it enables a more succinct encoding of properties (this follows from the results in [Wilke, 1999]). Still, many formulae of $\mathbf{ATL^+_{IR}}$ have their equivalent counterparts in **ATL**. For instance, the $\mathbf{ATL^+_{IR}}$ formula $\langle\langle jamesbond\rangle\rangle(\Box\neg\mathsf{crash} \wedge \Diamond\mathsf{land})$ can be equivalently rephrased in **ATL** as $\langle\langle jamesbond\rangle\rangle(\neg\mathsf{crash})\mathcal{U}(\mathsf{land} \wedge \langle\langle jamesbond\rangle\rangle\Box\neg\mathsf{crash})$.

In particular, we have that $\mathbf{ATL^+_{IR}}$ formulae can be equivalently translated into $\mathbf{ATL_{IR}}$ *with the "weak until" operator* [Harding et al., 2002]. We observe that in some cases the translation results in an exponential blowup of the length of the formula. Thus, $\mathbf{ATL^+_{IR}}$ has the same expressive power as "vanilla" $\mathbf{ATL_{IR}}$ with "weak until", but it seems to allow for exponentially more succinct and intuitive specifications of some properties (in a similar way to $\mathbf{CTL^+}$ vs. **CTL**, cf. [Wilke, 1999]).

In Section 5 we shall see that the more succinct language has its price: The model checking problem becomes computationally more expensive.

It is well known that the memoryless and perfect recall semantics for $\mathcal{L}_{ATL}$ formulae coincide [Alur et al., 2002; Schobbens, 2004] (cf. Theorem 2.24). The same is *not* true for $\mathcal{L}_{ATL^*}$, and in fact, already for $\mathcal{L}_{ATL^+}$. As a consequence,

$\mathcal{L}_{ATL^+}$ can be seen as the minimal well-known syntactic variant of the alternating time logics that distinguishes between the memoryless and perfect recall semantics.

We note that the *IR*- and *Ir*-semantics yield different validities for $\mathcal{L}_{ATL^+}$. For example, the formula

$$\langle\!\langle A\rangle\!\rangle(\Diamond \mathsf{p}_1 \wedge \Diamond \mathsf{p}_2) \leftrightarrow \langle\!\langle A\rangle\!\rangle\Diamond\big((\mathsf{p}_1 \wedge \langle\!\langle A\rangle\!\rangle\Diamond \mathsf{p}_2) \vee (\mathsf{p}_2 \wedge \langle\!\langle A\rangle\!\rangle\Diamond \mathsf{p}_1)\big)$$

is valid in the perfect recall semantics (*IR*), but not in the memoryless variant (*Ir*). Since $\mathbf{ATL_{IR}}$ and $\mathbf{ATL_{Ir}}$ have the same validities, $\mathcal{L}_{ATL^+}$ can also be seen as the minimal variant of the alternating time logics for which the *IR*- and *Ir*-semantics give rise to different *logics* in the traditional sense (as sets of valid sentences).

In conceptual terms, we can use $\mathcal{L}_{ATL^+}$ to specify a set of goals that should be achieved without saying in which order they should be accomplished, like in $\langle\!\langle robot\rangle\!\rangle(\Diamond \mathsf{cleanRoom} \wedge \Diamond \mathsf{packageDelivered})$. Moreover, $\mathcal{L}_{ATL^+}$ allows for reasoning about what can be achieved under certain assumptions about the agents' behaviour, as Example 2.36 shows. This kind of properties has been especially studied in deontic logic and normative systems (e.g., [Lomuscio and Sergot, 2003, 2004; Wozna and Lomuscio, 2004; van der Hoek et al., 2005b]), but also in reasoning about plausible behaviour of agents [Bulling and Jamroga, 2007a].

*Example 2.36.* Consider a class of systems, each represented by a concurrent game structure $\mathfrak{M}$ and a collection of *behavioural constraint sets* $\mathcal{B}_a$, one per agent $a \in \mathbb{A}\mathrm{gt}$. Like in [van der Hoek et al., 2005b], we define each behavioural constraint from $\mathcal{B}_a$ to be a pair $(q, \alpha)$, with the underlying interpretation that action $\alpha$ is forbidden for agent $a$ in state $q$ (for instance, by a social norm or law). Such representations can be reconstructed into a single CGS $\mathfrak{M}'$ by adding special propositions $\mathsf{V}_a$, $a \in \mathbb{A}\mathrm{gt}$, with the intuitive meaning "agent $a$ has committed a violation with its last action". If necessary, several copies of an original state $q$ from $\mathfrak{M}$ can be created, with different configurations of the $\mathsf{V}_a$ labels. Note that $\mathfrak{M}'$ is only linearly larger than $\mathfrak{M}$ wrt the number of original transitions in $\mathfrak{M}$.

Now, property "$a$ can enforce property $\gamma$ while complying with social norms" can be captured in $\mathfrak{M}'$ by the $\mathcal{L}_{ATL^+}$-formula $\langle\!\langle a\rangle\!\rangle((\Box\neg\mathsf{V}_a) \wedge \gamma)$. A similar property, "$b$ can enforce $\gamma$ provided that $a$ complies with norms" can be expressed with $\langle\!\langle b\rangle\!\rangle((\Box\neg\mathsf{V}_a) \rightarrow \gamma)$.

### 2.2.6 $\mathbf{EATL}^+$

Fairness conditions allow to focus on computations where no agent is neglected wrt given resources (e.g., access to power supply, processor time, etc.). Fairness is extremely important in asynchronous composition of agents. In general, it may happen that requests of a group $A \subset \mathbb{A}\mathrm{gt}$ are postponed forever in favour of actions from other agents. As a consequence, if we want to state any positive property about what $A$ can achieve, we need to refer explicitly to paths where $A$'s actions are always eventually executed. To this end, it is enough to augment $\mathcal{L}_{ATL^+}$ with the "always eventually" combination $\Box\Diamond$ as an additional primitive $\overset{\infty}{\Diamond}$.

**Definition 2.37 ($\mathcal{L}_{EATL^+}$).** *$\mathcal{L}_{EATL^+}(\Pi, \mathbb{A}\mathrm{gt})$ is a subset of $\mathcal{L}_{ATL^*}(\Pi, \mathbb{A}\mathrm{gt})$ obtained by extending $\mathcal{L}_{ATL^+}$-path formulae. The language is given as follows:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle\gamma, \text{ where } \gamma ::= \neg\gamma \mid \gamma \wedge \gamma \mid \bigcirc\varphi \mid \varphi\mathcal{U}\varphi \mid \overset{\infty}{\Diamond}\varphi.$$

We note that in $\mathcal{L}_{ATL^*}$, $\overset{\infty}{\Diamond}\varphi$ is expressible by $\Box\Diamond\varphi$ . Hence, we can use the ordinary $\mathcal{L}_{ATL^*}$-semantics to give truth to $\mathcal{L}_{EATL^+}$-formulae.

By the fact that **ECTL**$^+$ is more expressive than **CTL**$^+$ (which follows from **ECTL** being more expressive than **CTL** [Emerson, 1990]), we conjecture that $\mathbf{EATL}^+_{\mathbf{IR}}$ is also more expressive than $\mathbf{ATL}^+_{\mathbf{IR}}$. In particular, we conjecture that fairness constraints cannot be expressed in $\mathbf{ATL}^+_{\mathbf{IR}}$; that is, $\mathbf{EATL}^+_{\mathbf{IR}}$ is strictly more expressive than $\mathbf{ATL}^+_{\mathbf{IR}}$.

Hence the importance of $\mathbf{EATL}^+_{\mathbf{IR}}$ which allows for reasoning about the outcome of fair computations in model $\mathfrak{M}$. This is extremely important for the specification and verification of agents that act in an asynchronous environment. For example, most agent (and multi-agent) programming frameworks assume an asynchronous execution platform. In such settings, the following property from [Dastani and Jamroga, 2010] holds.

**Proposition 2.38 ([Dastani and Jamroga, 2010]).** *Let $\mathfrak{M}$ be a multi-agent program model, $q$ a state in $\mathfrak{M}$, and $\varphi$ an* **ATL** *formula. If there is a path in $\mathfrak{M}, q$ on which $\varphi$ never holds, then there must be an agent $i$ in $\mathfrak{M}$ so that, for each coalition $A \subseteq \mathbb{A}\mathrm{gt} \setminus \{i\}$, we have $\mathfrak{M} \not\models \langle\langle A\rangle\rangle\Diamond\varphi$.*

In other words, if the design of the program does not guarantee that $\varphi$ must eventually happen, then the execution platform (agent $i$ in the proposition above) can prevent actions of every coalition of "real" agents (from $\mathbb{A}\mathrm{gt} \setminus \{i\}$) and prevent them from achieving $\varphi$.

In $\mathbf{EATL}^+_{\mathbf{IR}}$, this can be overcome by putting fairness constraints explicitly in the formula. To make our discussion more concrete, let us assume that $\mathfrak{M}$ is an *asynchronous* CGSs as defined in [Alur et al., 2002]. That is, $\mathfrak{M}$ is a CGS where agent $k$ is designated as the *scheduler*. The scheduler's task is to choose the agent whose action is going to be executed, i.e., $d_k(q) = \mathbb{A}\mathrm{gt} \setminus \{k\}$ for every $q \in Q$, and for every pair of action profiles $\boldsymbol{\alpha}, \boldsymbol{\alpha}'$ that agree on the action of agent $j$ we have $o(q, \boldsymbol{\alpha}, j) = o(q, \boldsymbol{\alpha}', j)$. In our construction, we also assume that transitions by different agents lead to different states $(o(q, \boldsymbol{\alpha}, i) \neq$

$o(q, \boldsymbol{\alpha}', j)$ for $i \neq j$). Moreover, each state is labeled by proposition $\mathsf{act}_i$, where $i$ is the agent whose action was executed last.

Now, for example, the $\mathcal{L}_{EATL^+}$ formula

$$\langle\langle 1, 2\rangle\rangle\big((\overset{\infty}{\Diamond}\mathsf{act}_1 \wedge \overset{\infty}{\Diamond}\mathsf{act}_2) \rightarrow \Diamond\mathsf{cleanRoom}\big)$$

says that agents 1 and 2 can cooperate to make the room clean *for each course of events on which no agent is blocked forever.*

## 2.3 Further Strategic Epistemic Logics

In the previous section we have introduced extensions of ATLs by incomplete information (*iR*- and *ir*-semantics). The epistemic part was purely semantical and directly incorporated into the denotation of the cooperation modalities. There are more general attempts to logics combining strategic and epistemic concepts. The alternating time temporal epistemic logic **ATEL** from [van der Hoek and Wooldridge, 2003] extends **ATL** by standard knowledge operators. The logics *feasible* **ATEL** [Jonker, 2003], *uniform* **ATEL** [Jamroga, 2003] and *alternating time temporal observational logic* [Jamroga and van der Hoek, 2004] are of the same kind and overcome some problems encountered with **ATEL**. Similarly, *epistemic temporal strategic logic* [van Otterloo and Jonker, 2004] restricts to undominated strategies. Finally, the logic **CSL** (*constructive strategic logic*) has been proposed in [Jamroga and Ågotnes, 2006, 2007], an expressive logic that combines strategic and epistemic reasoning in a neat way. The latter comes for the cost of a non-standard semantics. In the following, we will present **ATEL** and **CSL** in more detail; the latter is used as the underlying logic of **CSLP** presented in Section 7.1. In that section we also point out the benefits of **CSL**. For a detailed discussion and a general overview of epistemic logics we refer to the original papers and to [Jamroga and Ågotnes, 2007; Jamroga and van der Hoek, 2004; Jamroga, 2004].

### 2.3.1 Alternating-Time Temporal Epistemic Logic: ATEL

The *alternating time temporal epistemic logic* **ATEL** [van der Hoek and Wooldridge, 2003] is a fusion of **ATL** with a standard **S5** epistemic logic together with group, distributed and common knowledge operators. The language of **ATEL** is given as follows:

$$\varphi ::= \psi \mid \mathsf{K}_a\varphi \mid \mathsf{C}_A\varphi \mid \mathsf{E}_A\varphi \text{ where } \psi \in \mathcal{L}_{ATL}$$

$a \in \mathbb{A}\text{gt}$ and $A \subseteq \mathbb{A}\text{gt}$. The epistemic operators have their standard meaning (from left to right): $a$ knows $\varphi$, group $A$ has common knowledge that $\varphi$, and everybody in $A$ knows that $\varphi$. In [van der Hoek and Wooldridge, 2003] alternating epistemic transitions systems were used to provide a semantics for the logic. Equivalently, we use ICGSs from Definition 2.27 (let $\mathfrak{M}$ be such a model) and extend the *Ir*-semantics from Definition 2.22 to give a meaning to the epistemic operators:

$$\mathfrak{M}, q \models_{Ir} \mathsf{K}_a\varphi \text{ iff for all } q' \in Q_{\mathfrak{M}} \text{ with } q \sim_a q' \text{ we have that } \mathfrak{M}, q' \models_{Ir} \varphi,$$
$$\mathfrak{M}, q \models_{Ir} \mathsf{E}_A\varphi \text{ iff for all } q' \in Q_{\mathfrak{M}} \text{ with } q \sim_A^E q' \text{ we have that } \mathfrak{M}, q' \models_{Ir} \varphi,$$
$$\mathfrak{M}, q \models_{Ir} \mathsf{C}_A\varphi \text{ iff for all } q' \in Q_{\mathfrak{M}} \text{ with } q \sim_A^C q' \text{ we have that } \mathfrak{M}, q' \models_{Ir} \varphi,$$

where $\sim_A^E := \bigcup_{a\in A} \sim_a$ and $\sim_A^C := (\sim_A^E)^*$. The latter denotes the reflexive and transitive closure of $\sim_A^E$. As mentioned in the literature, e.g. in [Jamroga, 2003; Jamroga and van der Hoek, 2004; Jonker, 2003], the logic yields some counterintuitive settings when it comes to the interplay of strategic ability and knowledge. The formula $\mathsf{K}_a\langle\langle a\rangle\rangle\varphi$, for instance, expresses that $a$ knows that it has a strategy to enforce $\varphi$. However, $a$ might *not* be able to identify a strategy since there can be a different winning strategy in each of the indistinguishable states. But the incomplete knowledge makes it impossible for the agent to identify (and thus execute) the correct strategy in each of these states.

In [Jonker, 2003] an extension of **ATEL** named *feasible* **ATEL** has been proposed in which additional modalities are introduced focussing on *uniform* strategies. We have introduced the notion *ir*-strategy to refer to the latter kind of strategies (cf. Definition 2.29).

### 2.3.2 Constructive Strategic Logic: CSL

In this section we present *constructive strategic logic* **CSL** [Jamroga and Ågotnes, 2006, 2007] which we will later extend by a concept of plausibility to reason about *rational* agents under incomplete information (see Chapter 7). On top of standard epistemic operators the language of **CSL** comes with *constructive* epistemic operators. The latter kind of operators model constructive knowledge of agents: Knowledge about the existence of a strategy does imply that the agents are also able to identify it.

**Definition 2.39 ($\mathcal{L}_{CSL}$).** *The logic $\mathcal{L}_{CSL}(\Pi, \mathbb{A}\mathrm{gt})$ is generated by the following grammar:*

$$\varphi ::= \psi \mid \mathbb{C}_A\varphi \mid \mathbb{E}_A\varphi \mid \mathbb{D}_A\varphi \textit{ where } \psi \in \mathcal{L}_{ATL}(\Pi, \mathbb{A}\mathrm{gt}).$$

Individual constructive knowledge operators are defined as $\mathbb{K}_a := \mathbb{C}_{\{a\}}$ and the standard epistemic operators (occurring, e.g., in **ATEL**) as

$$\mathsf{K}_a\varphi := \mathbb{K}_a\langle\langle\emptyset\rangle\rangle\varphi\,\mathcal{U}\varphi$$

and analogously for standard common ($\mathsf{C}_A$), mutual ($\mathsf{E}_A$), and distributed knowledge ($\mathsf{D}_A$) operators (cf. [Jamroga and Ågotnes, 2007] for details).

ICGSs from Definition 2.27 serve as models for **CSL**. Now we define the notion of *formula* $\varphi$ *being satisfied* by a (non-empty) set of states $Q'$ in model $\mathfrak{M}$, written $\mathfrak{M}, Q' \models \varphi$. We will also write $\mathfrak{M}, q \models \varphi$ as a shorthand for $\mathfrak{M}, \{q\} \models \varphi$. It is the latter notion of satisfaction (in single states) that we are ultimately interested in–but it is defined in terms of the (more general) satisfaction in sets of states. Let $\mathrm{img}(q, \mathcal{R})$ be the image of state $q$ with respect to binary relation $\mathcal{R}$, i.e., the set of all states $q'$ such that $q\mathcal{R}q'$. Moreover, we use $out(Q', s_A)$ as a shorthand for $\bigcup_{q \in Q'} out(q, s_A)$ (cf. Definition 2.21), and $\mathrm{img}(Q', \mathcal{R})$ as a shorthand for $\bigcup_{q \in Q'} \mathrm{img}(q, \mathcal{R})$.

**Definition 2.40 (Semantics, CSL).** *Let* $\mathfrak{M}$ *be an* ICGS. *The semantics for* $\mathcal{L}_{CSL}$*, denoted by* $\models_{\mathsf{CSL}}$*, is given by the following clauses:*

$\mathfrak{M}, Q' \models_{\mathsf{CSL}} \mathsf{p}$ *iff* $\mathsf{p} \in \pi(q)$ *for each* $q \in Q'$*;*
$\mathfrak{M}, Q' \models_{\mathsf{CSL}} \neg\varphi$ *iff* $\mathfrak{M}, Q' \not\models_{\mathsf{CSL}} \varphi$*;*
$\mathfrak{M}, Q' \models_{\mathsf{CSL}} \varphi \wedge \psi$ *iff* $\mathfrak{M}, Q' \models_{\mathsf{CSL}} \varphi$ *and* $\mathfrak{M}, Q' \models_{\mathsf{CSL}} \psi$*;*
$\mathfrak{M}, Q' \models_{\mathsf{CSL}} \langle\langle A \rangle\rangle \bigcirc \varphi$ *iff there exists an ir-strategy* $s_A \in \Sigma_A^{ir}$ *such that, for each* $\lambda \in out(Q', s_A)$*, we have that* $\mathfrak{M}, \{\lambda[1]\} \models_{\mathsf{CSL}} \varphi$*;*
$\mathfrak{M}, Q' \models_{\mathsf{CSL}} \langle\langle A \rangle\rangle \Box \varphi$ *iff there exists an ir-strategy* $s_A \in \Sigma_A^{ir}$ *such that, for each* $\lambda \in out(Q', s_A)$ *and* $i \in \mathbb{N}_0$*, we have* $\mathfrak{M}, \{\lambda[i]\} \models_{\mathsf{CSL}} \varphi$*;*
$\mathfrak{M}, Q' \models_{\mathsf{CSL}} \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$ *iff there exists an ir-strategy* $s_A \in \Sigma_A^{ir}$ *such that, for each* $\lambda \in out(Q', s_A)$*, there is an* $i \in \mathbb{N}_0$ *for which* $\mathfrak{M}, \{\lambda[i]\} \models_{\mathsf{CSL}} \psi$ *and* $\mathfrak{M}, \{\lambda[j]\} \models_{\mathsf{CSL}} \varphi$ *for each* $0 \leq j < i$*.*
$\mathfrak{M}, Q' \models_{\mathsf{CSL}} \hat{\mathcal{K}}_A \varphi$ *iff* $\mathfrak{M}, \mathrm{img}(Q', \sim_A^{\mathcal{K}}) \models_{\mathsf{CSL}} \varphi$ *(where* $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ *and* $\mathcal{K} = C, E, D$*, respectively).*

*where relations* $\sim_A^E$*,* $\sim_A^C$ *and* $\sim_A^D$*, used to model group epistemics, are derived from the individual relations of agents from* $A$*. First,* $\sim_A^E$ *is the union of relations* $\sim_a$*,* $a \in A$*. Next,* $\sim_A^C$ *is defined as the transitive closure of* $\sim_A^E$*. Finally,* $\sim_A^D$ *is the intersection of all the* $\sim_a$*,* $a \in A$*.*

The macros of the standard epistemic operators may seem awkward but they actually meet the expected semantics as illustrated below:

$\mathfrak{M}, Q' \models_{\mathsf{CSL}} \mathsf{K}_a \varphi$
iff $\mathfrak{M}, \mathrm{img}(Q', \sim_{\{a\}}^C) \models_{\mathsf{CSL}} \langle\langle \emptyset \rangle\rangle \varphi \mathcal{U} \varphi$
iff for each $\lambda \in out(\mathrm{img}(Q', \sim_{\{a\}}^C), \emptyset)$, $\mathfrak{M}, \lambda[0] \models_{\mathsf{CSL}} \varphi$
iff for each $q \in \mathrm{img}(Q', \sim_{\{a\}}^C)$, $\mathfrak{M}, q \models_{\mathsf{CSL}} \varphi$.

Due to the non-standard semantics, knowledge is an *KD45*-modality; the truth axiom $T$ is not valid. This issue is discussed in more detail in [Jamroga and Ågotnes, 2007].

# 3

# Relating Games and Logics

## Contents



In this chapter we analyse the connection between game theory and strategic logics. We show how these logics can be used to characterise game-theoretic solution concepts. In Part II of this thesis we use descriptions of the latter kind to model and to reason about agents' rational behaviour.

## 3.1 Concepts From Game Theory

The following introduction to game theory is taken from [Bulling et al., 2009b] which in turn is mostly based on [Osborne and Rubinstein, 1994].

We start with the definition of a *normal form* game, also called *strategic game*. We follow the terminology of [Osborne and Rubinstein, 1994].

**Definition 3.1 (Normal Form (NF) Game, $a_i$, $a_{-i}$).** *A (perfect information)* normal form game *$\Gamma$, is a tuple of the form $\Gamma = \langle \mathcal{P}, \mathcal{A}_1, \ldots, \mathcal{A}_k, \mu \rangle$, where*

| 1\ 2 | $a_2^1$ | $a_2^2$ |
|---|---|---|
| $a_1^1$ | $\langle \boldsymbol{\mu}_1(a_1^1, a_2^1), \boldsymbol{\mu}_2(a_1^1, a_2^1)\rangle$ | $\langle \boldsymbol{\mu}_1(a_1^1, a_2^2), \boldsymbol{\mu}_2(a_1^1, a_2^2)\rangle$ |
| $a_1^2$ | $\langle \boldsymbol{\mu}_1(a_1^2, a_2^2), \boldsymbol{\mu}_2(a_1^2, a_2^2)\rangle$ | $\langle \boldsymbol{\mu}_1(a_1^2, a_2^1), \boldsymbol{\mu}_2(a_1^2, a_2^2)\rangle$ |

**Fig. 3.1.** Payoff matrix for 2 players and $2 \times 2$ strategies

| 1\ 2 | Head | Tail |
|---|---|---|
| Head | (1, -1) | (-1, 1) |
| Tail | (-1, 1) | (1, -1) |

| 1\ 2 | C | D |
|---|---|---|
| C | (3, 3) | (0, 5) |
| D | (5, 0) | **(1, 1)** |

| 1\ 2 | Dove | Hawk |
|---|---|---|
| Dove | (3, 3) | **(1, 4)** |
| Hawk | **(4, 1)** | (0, 0) |

**Fig. 3.2.** Payoff matrices for Matching Pennies, Prisoner's Dilemma, and Hawk-Dove. Nash equilibria are set in bold font.

- $\mathcal{P}$ *is a finite set of* players *(or* agents*), with* $|\mathcal{P}| = k$*,*
- $\mathcal{A}_i$ *are nonempty sets of* actions *(or* strategies*) for player* $i$*,*
- $\mu : \mathcal{P} \to (\times_{i=1}^{k} \mathcal{A}_i \to \mathbb{R})$ *is the* payoff function *(which we also write* $\langle \mu_1, \ldots, \mu_k \rangle$*).*

*Combinations of actions (resp. strategies, payoffs), one per player, will be called* action profile *(resp.* strategy profile, payoff profile*). Given a strategy profile* $a \in \times_{i=1}^{k} \mathcal{A}_i$ *we write* $a_i$ *(resp.* $a_{-i}$*) to denote* $i$*'s strategy* $a_i$ *(resp. the strategy profile* $(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k)$ *of all players apart from* $i$*). Moreover, we use* $(a_i, a'_{-i})$ *to refer to the complete strategy profile resulting from combining the strategies* $a_{-i}$ *with the strategy* $a_i$*.*

Such games are usually depicted as payoff matrices. For example, a game with 2 players having 2 strategies each is represented by the matrix in Figure 3.1.

*Example 3.2 (Classical NF Games).* Some classical NF games with 2 players and 2 strategies are shown in Figure 3.2. In the *Matching Pennies* game, player 1 wins when both pennies show the same side. Otherwise player 2 wins. In the *Prisoner's Dilemma*, two prisoners can either cooperate or defect with the police. Finally, the *Hawk-Dove* game is similar, but the payoffs are different. The higher the payoff the better it is for the respective player.

**Definition 3.3 (Solution concepts in games).** *There are several well-known* solution concepts *such as follows.*

**Best Response (BR):** *A strategy* $a_i$ *is a* best response of $i$ against $a_{-i}$ *if it is among the best strategy* $i$ *can chose if the opponents play* $a_{-i}$*; i.e.*

$$\forall a'_i \in \mathcal{A}_i \ (\mu_i((a_{-i}, a_i)) \geq \mu_i((a_{-i}, a'_i))).$$

**Nash Equilibrium (NE):** *A strategy profile such that no agent can unilaterally deviate from its strategy and get a better payoff; i.e., a profile* $a^*$ *is a* NE *if for each player* $i$, $a_i^*$ *is a best response to* $a_{-i}^*$.

**Pareto Optimality (PO):** *There is no other strategy profile that leads to a payoff profile which is at least as good for each agent, and strictly better for at least one agent; i.e.,* $a^*$ *is* PO *if there is no profile* $a'$ *such that for all players* $i$, $\mu_i(a') \geq \mu_i(a^*)$ *and for some player* $j$, $\mu_j(a') > \mu_j(a^*)$.

**Weakly Undominated Strategies (UNDOM):** *These are strategies that are not dominated by any other strategy, i.e.,* $a_i$ *is weakly undominated for* $i$ *if there is no strategy* $a_i'$ *at least as good for all the responses of the opponent, and strictly better for at least one response.*

*We point out that some solution concepts yield sets of individual strategies (UNDOM), while others produce rather sets of strategy* profiles *(NE, PO).*

In the examples from Figure 3.2, there is no Nash equilibrium for the Matching Pennies game, exactly one Nash equilibrium for the Prisoner's Dilemma (namely, the strategy profile $\langle D, D\rangle$), and two Nash equilibria for the Hawk-Dove game ($\langle Hawk, Dove\rangle$ and $\langle Dove, Hawk\rangle$).

In NF games, agents do their moves *simultaneously*: They do not see the move of the opponents and therefore cannot act accordingly. On the other hand, there are many games where the move of one player should depend on the preceding move of the opponent, or even on the *whole history*. This idea is captured in games of *extensive form*.

**Definition 3.4 (Extensive Form (EF) Game).** *A (perfect information)* extensive (form) game $\Gamma$ *is a tuple of the form* $\Gamma = \langle \mathcal{P}, \mathcal{A}, H, ow, u\rangle$, *where:*

- $\mathcal{P}$ *is a finite set of players,*
- $\mathcal{A}$ *a finite set of actions (moves),*
- $H$ *is a set of finite action sequences (game histories), such that (1)* $\emptyset \in H$, *(2) if* $h \in H$, *then every initial segment of* $h$ *is also in* $H$. *We use the notation* $\mathcal{A}(h) = \{m \mid h \circ m \in H\}$ *to denote the* moves available at $h$, *and we define* $Term = \{h \mid \mathcal{A}(h) = \emptyset\}$, *the set of* terminal *positions,*
- $ow : H \to \mathcal{P}$ *defines which player "owns" history* $h$, *i.e., has the next move given* $h$,
- $u : \mathcal{P} \times Term \to U$ *assigns agents' utilities to every terminal position of the game.*

*We will usually assume that the set of utilities* $U$ *is finite.*

Such games can be easily represented as trees of all possible plays.

*Example 3.5 (Bargaining).* Consider bargaining with discount [Osborne and Rubinstein, 1994; Sandholm, 1999]. Two players, 1 and 2, bargain about how

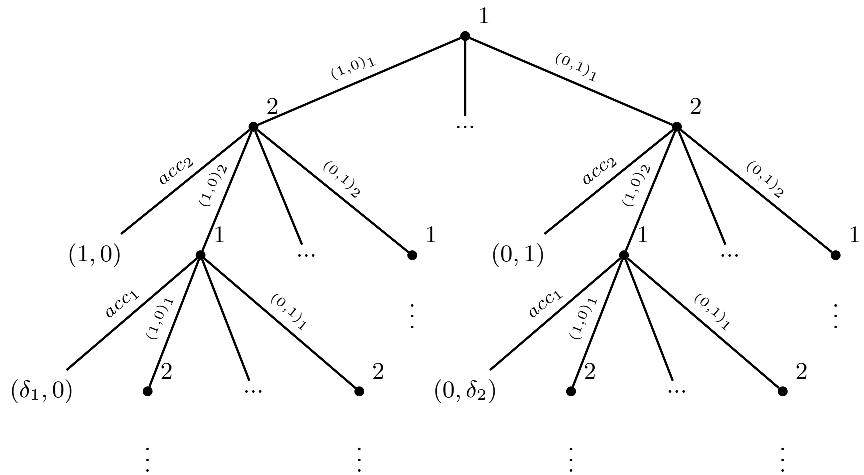



**Fig. 3.3.** The bargaining game.

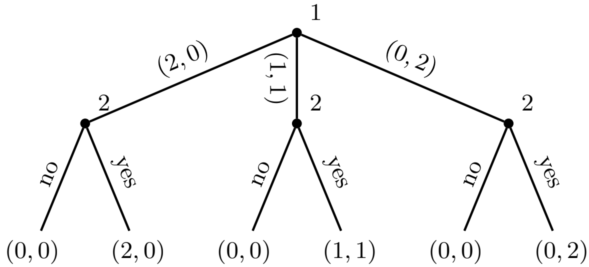
to split goods worth initially $w_0 = 1$ EUR. After each round without agreement, the subjective worth of the goods reduces by *discount rates* $\delta_1$ (for player $a_1$) and $\delta_2$ (for player $a_2$). So, after $t$ rounds, the goods are worth $\langle \delta_1^t, \delta_2^t \rangle$, respectively. Subsequently, $a_1$ (if $t$ is even) or $a_2$ (if $t$ is odd) makes an offer to split the goods in proportions $\langle x, 1-x \rangle$, and the other player accepts or rejects it. If the offer is accepted, then $a_1$ takes $x\delta_1^t$, and $a_2$ gets $(1-x)\delta_2^t$; otherwise the game continues. The (infinite) extensive form game is shown in Figure 3.3. Note that the tree has infinite depth as well as an infinite branching factor.

In order to obtain a finite set of payoffs, it is enough to assume that the goods are split with finite precision represented by a rounding function $r : \mathbb{R} \to \mathbb{R}$. So, after $t$ rounds, the goods are in fact worth $\langle r(\delta_1^t), r(\delta_2^t) \rangle$, respectively, and if the offer is accepted, then $a_1$ takes $r(x\delta_1^t)$, and $a_2$ gets $r((1-x)\delta_2^t)$.

A *strategy* for player $i \in \mathcal{P}$ in extensive game $\Gamma$ is a function that assigns a legal move to *each* history owned by $i$. A *strategy profile* (i.e., a combination of strategies, one per player) determines a unique path from the game root ($\emptyset$) to one of the terminal nodes (and hence also a single profile of payoffs). As a consequence, one can construct the corresponding normal form game $NF(\Gamma)$ by enumerating strategy profiles and filling the payoff matrix with resulting payoffs.

*Example 3.6 (Sharing Game).* Consider the *Sharing Game* in Figure 3.4A. Its corresponding normal form game is presented in Figure 3.4B. Firstly, player

(A)




(B)

| 1\ 2 | nnn | nny | nyn | nyy | ynn | yny | yyn | yyy |
|---|---|---|---|---|---|---|---|---|
| (2, 0) | **(0, 0)** | **(0, 0)** | (0, 0) | (0, 0) | **(2, 0)** | **(2, 0)** | **(2, 0)** | **(2, 0)** |
| (1, 1) | (0, 0) | (0, 0) | **(1, 1)** | **(1, 1)** | (0, 0) | (0, 0) | (1, 1) | (1, 1) |
| (0, 2) | (0, 0) | **(0, 2)** | (0, 0) | (0, 2) | (0, 0) | (0, 2) | (0, 0) | (0, 2) |

**Fig. 3.4.** The Sharing game: (A) Extensive form; (B) Normal form. Nash equilibria are set in bold font. A strategy $abc$ ($a, b, c \in \{y, n\}$) of player 2 denotes the strategy in which 2 plays $a$ (resp. $b$, $c$) if player 1 has played $(2, 0)$ (resp. $(1, 1)$, $(0, 2)$) where $n$ refers to "no" and $y$ to "yes".

1 can suggest how to share, say, two 1 EUR coins. E.g. $(2, 0)$ means that 1 gets two EUR and 2 gets nothing. Subsequently, player 2 can accept the offer or reject it; in the latter case both players get nothing.

The game includes 3 strategies for player 1 (which can be denoted by the action that they prescribe at the beginning of the game), and 8 strategies for player 2 (generated by the combination of actions prescribed for the second move), which gives 24 strategy profiles in total. However, not all of them seem plausible. Constraining the possible plays to Nash equilibria only, we obtain 9 "rational" strategy profiles (cf. Figure 3.4B), although it is still disputable if all of them really "make sense".

A *subgame* of an extensive game $\Gamma$ is defined by a subtree of the game tree of $\Gamma$.

**Definition 3.7 (Subgame Perfect Nash Equilibrium (SPN)).** *This solution concept is an extension of NE: A strategy is a SPN in $\Gamma$ if it is a NE in $\Gamma$ and, in addition, a NE in all subgames of $\Gamma$.*

*Example 3.8 (Sharing Game ctd.).* Consider again the game from Example 3.6. While the game has 9 Nash equilibria, only two of them are subgame perfect ($\langle (2, 0), yyy\rangle$ and $\langle (1, 1), nyn\rangle$).

*Example 3.9 (Bargaining ctd.).* We consider the bargaining game from Example 3.12. The game has an immense number of possible outcomes. Still worse, every strategy profile

$$s^x : \begin{cases} a_1 \text{ always offers } \langle x, 1-x \rangle \text{, and agrees to } \langle y, 1-y \rangle \text{ for } y \geq x \\ a_2 \text{ always offers } \langle x, 1-x \rangle \text{, and agrees to } \langle y, 1-y \rangle \text{ iff } 1-y \geq 1-x \end{cases}$$

is a Nash equilibrium (NE): *An agreement is reached in the first round.* Thus, each split $\langle x, 1-x \rangle$ can be achieved through a Nash equilibrium; it seems that a stronger solution concept is needed. Indeed, the game has a unique *subgame perfect Nash equilibrium.* Because of the finite precision, there is a minimal round $T$ with $r(\delta_i^{T+1}) = 0$ for $i = 1$ or $i = 2$. For simplicity, assume that $i = 2$ and agent $a_1$ is the offerer in $T$ (i.e., $T$ is even). Then, the only subgame perfect NE is given by the strategy profile $s^\kappa$ with $\kappa = (1-\delta_2)\frac{1-(\delta_1\delta_2)^{\frac{T}{2}}}{1-\delta_1\delta_2} + (\delta_1\delta_2)^{\frac{T}{2}}$. The goods are split $\langle \kappa, 1-\kappa \rangle$; the agreement is reached in the first round (cf. Section A.4).[1]

## 3.2 Reasoning about Games

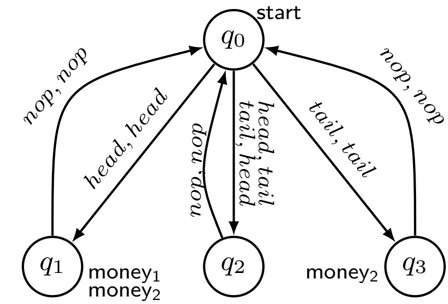
In this section we present some important ideas that form the starting point for later sections when analysing the behaviour of rational agents (cf. Chapter 6 and 7). We discuss informally how the notion of strategic ability in **ATL** can be refined so that it takes into account only "sensible" behaviour of agents and we summarise a correspondence between extensive games and CGSs, the models of ATLs. Subsequently, we present two logics that can be used to implement these ideas from game theory directly. The first logic *Game Logic with preferences* **GLP** [van der Hoek et al., 2004] does so in a limited way. The other extension of **ATL**, called **ATLI** ("**ATL** with Intentions") [Jamroga et al., 2005], is more general and will later serve as an intermediate logical framework and as a motivation for our logic **ATLP** defined in Section 6.1. Finally, we demonstrate how several game-theoretical solution concepts can be expressed in **ATLI**. For this purpose we introduce *general* or *qualitative* solution concepts, where **ATL** path formulae are used to define the winning conditions, instead of utilities.

### 3.2.1 ATL and Rational Play

We begin this section with an example.

[1] For the standard version of bargaining with discount (with the continuous set of payoffs $[0, 1]$), cf. [Osborne and Rubinstein, 1994; Sandholm, 1999]. Restricting

(A)




(B)

| $1\backslash 2$ | $s_h$ | $s_t$ |
|---|---|---|
| $s_h$ | $1, 1$ | $0, 0$ |
| $s_t$ | $0, 0$ | $0, 1$ |

**Fig. 3.5.** *Asymmetric matching pennies*: **(A)** Concurrent game structure $\mathfrak{M}_1$. In $q_0$ the agents can choose to show *head* or *tail*. Both agents can only execute action *nop* (no operation) in states $q_1, q_2, q_3$. **(B)** The corresponding NF game. We use $s_h$ (resp. $s_t$) to denote the strategy in which the player always shows head (resp. tail) in $q_0$ and *nop* in $q_1$, $q_2$, and $q_3$.

*Example 3.10 (Asymmetric matching pennies).* Consider a variant of the matching pennies game, presented in Figure 3.5A. Formally, the model is given as follows:

$$\mathfrak{M}_1 = \langle \{1, 2\}, \{q_0, q_1, q_2, q_3\}, \{\mathsf{start}, \mathsf{money}_1, \mathsf{money}_2\}, \pi, \{head, tail, nop\}, d, o\rangle$$

where $\pi$ is defined as in the picture ($\pi(q_0) = \{\mathsf{start}\}$ etc.), $d(a, q_0) = \{head, tail\}$ for $a = 1, 2$, and $d(a, q) = \{nop\}$ for $a = 1, 2$ and $q = q_1, q_2, q_3$. The transition function $o$ can also be read off from the picture. We use *nop* (no operation) as a "default" action in states $q_1$, $q_2$, and $q_3$ that brings the system back to the initial state. The intuition is that the game is played ad infinitum. Alternatively, one might add loops to states $q_1, q_2$ and $q_3$ to model a game that is played only once.

If both players show heads in $q_0$, both win a prize in the next step; if they both show tails, only player 2 wins. If they show different sides, nobody wins. Note that, e.g., $\mathfrak{M}_1, q_0 \models \langle\!\langle 2 \rangle\!\rangle \Box \neg \mathsf{money}_1$, because agent 2 can play *tail* all the time, preventing 1 from winning the prize. On the other hand, $\mathfrak{M}_1, q_0 \models \neg\langle\!\langle 2 \rangle\!\rangle \Diamond \mathsf{money}_2$: Agent 2 has no strategy to guarantee that it will win.

The CGS in Figure 3.5A determines the set of available strategy profiles. However, it does not say anything about players' preferences. Suppose now that the players are only interested in getting some money sometime in the

---

the payoffs to a finite set requires to alter the solution slightly [Ståhl, 1972; Mas-Colell et al., 1995], see also Appendix A.4.

future (but it does not matter when and/or how much). The corresponding normal form game under this assumption is depicted in Figure 3.5B.

Such an analysis of the game is of course correct, yet it appears to be quite coarse. It seems natural to assume that players prefer winning money over losing it. If we additionally assume that the players are rational thinkers, it seems plausible that player 1 should always play head, as it keeps the possibility of getting money open (while playing tail guarantees loss). Under this assumption, player 2 has complete control over the outcome of the game: It can play head too, granting itself and the other agent with the prize, or respond with tail, in which case both players lose. This kind of analysis corresponds to the game-theoretical notion of *weakly dominant strategy*: For agent 1, playing head is dominant in the corresponding normal form game in Figure 3.5B, while both strategies of player 2 are undominated, so they can be in principle considered for playing.

It is still possible to refine our analysis of the game. Note that 2, knowing that 1 ought to play head and preferring to win money too, should decide to play *head* herself. This kind of reasoning corresponds to the notion of *iterated undominated strategies*. If we assume that both players do reason this way, then $\langle s_h, s_h \rangle$ is the only rational strategy profile, and the game should end with both agents winning the prize.

### 3.2.2 CGSs vs. Extensive Games

In this section we recall the correspondence between extensive form games and the semantical models of **ATL**, proposed in [Jamroga et al., 2005] and inspired by [Baltag, 2002; van der Hoek et al., 2005a].

We recall after [Baltag, 2002; Jamroga et al., 2005] that CGSs embed *extensive form games with perfect information* in a natural way. This can be done, e.g., by adding auxiliary propositions to CGSs, that describe the payoffs of agents. With this perspective, concurrent game structures can be seen as a strict generalisation of extensive form games.

We only consider game trees in which the set of payoffs is finite. Let $U$ denote the set of all possible utility values in a game; $U$ will be finite and fixed for any given game. For each value $v \in U$ and agent $a \in \mathbb{A}\mathrm{gt}$, we introduce a proposition $\mathsf{p}_\mathsf{a}^\mathsf{v}$ into our set $\Pi$, and fix $\mathsf{p}_\mathsf{a}^\mathsf{v} \in \pi(q)$ iff $a$ gets payoff of at least $v$ in $q$.[2] States in the model represent finite histories in the game. In particular, we us $\emptyset$ to denote the root of the game. The correspondence between an extensive game $\Gamma$ and a CGS $\mathfrak{M}$ can be captured as follows.

**Definition 3.11 (From extensive form games to CGSs).** *We say that a* CGS $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o \rangle$ corresponds to *an extensive form game* $\Gamma = \langle \mathcal{P}, \mathcal{A}, H, ow, u \rangle$ *if, and only if, the following holds:*

[2] Note that a state labeled by $\mathsf{p}_\mathsf{a}^\mathsf{v}$ is also labeled by $\mathsf{p}_\mathsf{a}^{\mathsf{v}'}$ for all $v' \in U$ where $v' < v$.

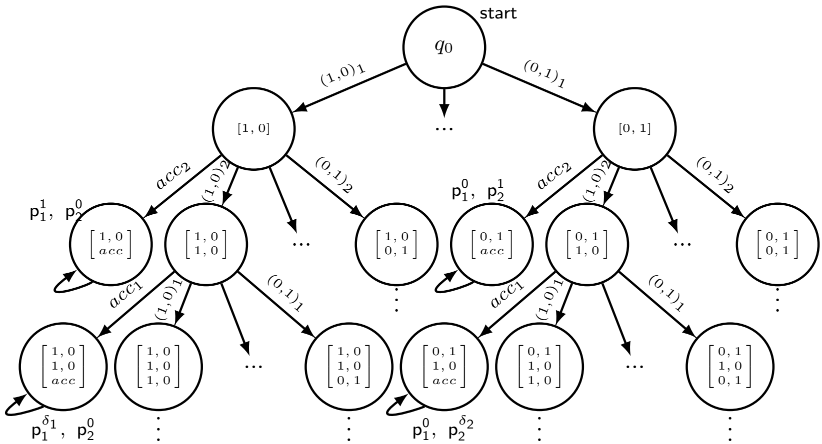



**Fig. 3.6.** CGS $\mathfrak{M}_2$ for the bargaining game

- $\mathbb{A}\mathrm{gt} = \mathcal{P}$,
- $Q = H$,
- *$\Pi$ and $\pi$ include propositions $\mathsf{p}_a^v$ to emulate utilities for terminal states in the way described above,*
- $Act = \mathcal{A} \cup \{nop\}$,
- $d_a(q) = \mathcal{A}(q)$ *if* $a = ow(q)$ *and* $d_a(q) = \{nop\}$ *otherwise,*
- $o(q, nop, \ldots, m, \ldots, nop) = q \cdot m$, *and*
- $o(q, nop, nop, \ldots, nop) = q$ *for* $q \in Term$.

*We use $\mathfrak{M}(\Gamma)$ to refer to the* CGS *which corresponds to $\Gamma$.*

*Example 3.12 (Bargaining in a* CGS*).* We consider the bargaining game from Example 3.5, but this time as a CGS. The CGS corresponding to the game is shown in Figure 3.6. Nodes represent various states of the negotiation process, and arcs show how agents' moves change the state of the game. A node label refers to the history of the game for better readability. For instance, $\begin{bmatrix} 0,1 \\ 1,0 \\ acc \end{bmatrix}$ has the meaning that in the first round 1 offered $\langle 0, 1\rangle$ which was rejected by 2. In the next round 2's offer $\langle 1, 0\rangle$ has been accepted by 1 and the game has ended.

Note that, for each extensive form game $\Gamma$, there is a corresponding CGS, but the reverse is not true: Concurrent game structures can include cycles and simultaneous moves of players, which are absent in game trees. For those CGSs that correspond to some extensive form game, we get an implicit correspondence to a normal form game. We will extend this notion of correspondence to all CGSs in Section 3.3.2.

### 3.2.3 A Modal Logic for Games

In [Harrenstein et al., 2003] a modal logic for characterising solution concepts is presented. The main construct of the logic is $[\beta]\varphi$ where $\beta$ ranges over preference relations, and complete and partial strategy profiles. The three kinds of operators have the following meaning where $pref_i$ (resp. $\sigma$ and $i$) represents the preference relation of player $i$ (resp. a complete strategy profile, a player):

$[pref_i]\varphi$: $\varphi$ holds in all states at least as preferable to player $i$ as the current one.
$[\sigma]\varphi$: $\varphi$ will hold in the final state reached if all players follow $\sigma$.
$[\sigma_{-i}]\varphi$: $\varphi$ will hold in all final states reached if all players apart from $i$ follow $\sigma$.

Note, that the modal logic is built on a great number of modal operators. In the paper it is shown how solution concepts can be described by formulae of this logic. For instance, the formulae

$$(\neg[\sigma_{-i}]\neg[pref_i]\varphi) \rightarrow [\sigma]\varphi$$

expresses that $\sigma_i$ is a *best response* to $\sigma_{-i}$ with respect to $\varphi$: If there is a strategy of $i$ (note that $\sigma_{-i}$ does not fix a strategy for $i$) such that the state reachable satisfies $\varphi$ and is among the most preferred ones regarding $i$ then the strategy $\sigma_i$ (which is included in $\sigma$) does also bring about $\varphi$.

A complete axiomatisation is also presented. The logic is a very special-purpose logic; strategies and preferences are first-class citizens. The main task of the logic is to reason about the outcome of extensive form games; in principal, these are also taken as models for the logic. So, it is not possible to reason about temporal behaviours of players. The logic **ATLP** presented in Section 6.1 overcomes this limitation.

### 3.2.4 Game Logic with Preferences

Game logic with preferences [van der Hoek et al., 2004] (**GLP**) is, to our knowledge, the only logic designed to address the outcome of rational play in games with perfect information. Here, we briefly summarise the idea.

The central idea of **GLP** is facilitated by the *preference operator* $[A : \varphi]$. The interpretation of $[A : \varphi]\psi$ in model $\mathfrak{M}$ is given as follows: If the truth of $\varphi$ can be enforced by group $A$, then we remove from the model all the actions of $a$ that do *not* enforce it and evaluate $\psi$ in the resulting model. Thus, the evaluation of **GLP** formulae is underpinned by the assumption that *rational agents satisfy their preferences whenever they can.* The requirement applies to all the subtrees of the game tree, and is called "subgame perfectness" by the authors. Formulae of **GLP** are defined by

$$\varphi ::= \varphi_0 \mid \varphi \vee \varphi \mid \neg\varphi \mid [A : \varphi_0]\varphi$$

where $\varphi_0$ is a propositional formula over some set $\Pi$ of proposition and $A \subseteq \mathbb{A}\mathrm{gt}$ a group of agents. Models of **GLP** are essentially *finite* perfect information extensive form games (without utility functions) extended with a labelling function assigning propositions to final nodes of histories. Formally, let $\langle \mathcal{P}, \mathcal{A}, H, ow, u\rangle$ be a perfect information extensive form game (cf. Definition 3.4) and let $\pi : Z(H) \to \mathcal{P}(\Pi)$ assign to each final history a set of propositions true at it. $Z(H)$ denotes the set of finite histories of $H$. Then, a **GLP**-model is given by

$$\Gamma = (\mathcal{P}, \mathcal{A}, H, ow, \Pi, \pi).$$

Propositional formulae $\varphi_0$ are interpreted over $\pi$ and final histories $h \in Z(H)$ as usual; we write $\pi, h \models \varphi_0$ if $\varphi_0$ is true with respect to $\pi$ and $h$. The other formulae are interpreted as stated below:

$\Gamma \models \Box\varphi_0$ iff for all $h \in Z(H) : \pi, h \models \varphi_0$,
$\Gamma \models \varphi \vee \psi$ iff $\Gamma \models \varphi$ or $\Gamma \models \psi$,
$\Gamma \models \neg\varphi$ iff $\Gamma \not\models \varphi$,
$\Gamma \models [A : \varphi_0]\varphi$ iff $Up(\Gamma, A, \varphi_0) \models \psi$.

The model update operator $Up$ removes from the model all actions not ensuring $\varphi_0$ if there is a way to ensure $\varphi_0$. More precisely, it takes the most general *choice strategy* $s_A$ of players $A$ such that for each subgame $\Gamma'$ of $\Gamma$ all final histories consistent with $s_A$ satisfy $\varphi_0$ if such a strategy exists. A *choice strategy* is a generalisation of a strategy in extensive form games. Formally, a choice strategy for $a$ is a function assigning possibly more than one action to each history owned by $a$; that is, it is a function (cf. Definition 3.4)

$$s_a : \{h \in H \mid ow(h) = a\} \to \mathcal{P}(\mathcal{A}) \text{ with } s_a(h) \subseteq \mathcal{A}(h).$$

As before, a *collective* choice strategy is a set of individual choice strategies. Such a choice strategy can be considered as the union of strategies. More formally, $Up(\Gamma, A, \varphi_0)$ is the game that results if $\Gamma$ is restricted according to choice strategy $s_A^*$ which is defined as follows. $s_A^*$ is the most general choice strategy such that if any subgame $\Gamma'$ of $\Gamma$ in which $A$ has a choice strategy $s'_A$ such that the restriction of $\Gamma'$ according to $s'_A$ satisfies $\Box\varphi_0$, then the strategy $s_A^*$ enforces $\Box\varphi_0$ in $\Gamma'$. Intuitively, $s_A^*$ removes a minimal number of actions in each subgame such that $\Box\varphi_0$ if there is a way to ensure $\varphi_0$. We note that if $\varphi_0$ cannot be ensured in a subgame no action is removed. We call $s_A^*$ the *subgame-perfect strategy* of $A$ for $\varphi_0$ in $\Gamma$ and denote it by

$$s^*_A = s^*(A, \Gamma, \varphi_0).$$

The scope of **GLP**, however, is limited for several reasons. Firstly, the models of **GLP** are restricted to finite game trees (in the sense that propositional formulae can only be evaluated at finite histories). Secondly, agents' preferences must be specified with propositional (non-modal) formulae, and they are evaluated only at the terminal states of the game. The temporal part of the language is limited, too. Lastly, and perhaps most importantly, the semantics of **GLP** is based on a very specific notion of rationality (see above). One can easily imagine variants of the semantics, in which other rationality criteria are used (Nash equilibria, Pareto optimal strategies, undominated strategies, etc.) to eliminate "irrational" strategies. The strategies used seem to be quite strong either there is such a "very good" strategy or not. Indeed, a preliminary version of **GLP** was based on the notion of Nash equilibrium rather than "subgame perfectness" [van Otterloo et al., 2004]. In this thesis, we want to allow as much flexibility as possible with respect to the choice of a suitable solution concept.

In Section 6.3 we show that the logic **ATLP** embeds **GLP** by just plugging-in this very notion of rationality.

### 3.2.5 ATLI: ATL with Intentions

The correspondence between extensive form games and concurrent game structures gives us a way of performing game-theoretical analyses on the latter. In particular, game-theoretical solution concepts become meaningful for these CGSs. In this section we present the logic **ATLI** [Jamroga et al., 2005] that allows to describe several important notions of rationality from game theory. We will later show how these characterisations can be "plugged" into the new logic **ATLP** introduced in Chapter 6 so that one can reason about the outcome of rational play in a precisely defined sense.

We also point out after [Jamroga et al., 2005] that these characterisations give rise to generalised versions of solution concepts which can be applied to *all* CGSs, and not only to those that correspond to some extensive form game. These solution concepts are also more flexible in describing winning criteria to agents.

*Alternating time temporal logic with intentions* (**ATLI**) extends **ATL** with formulae $(\mathbf{str}_a\sigma_a)\varphi$ with the intuitive reading: *Suppose that player $a$ intends to play according to strategy $\sigma_a$, then $\varphi$ holds.* Thus, it allows to refer to agents' strategies explicitly via terms $\sigma_a$. Let $\mathfrak{Str} = \bigcup_{a\in\mathbb{A}\mathrm{gt}} \mathfrak{Str}_a$ be a finite set of *strategic terms*. $\mathfrak{Str}_a$ is used to denote *individual strategies* of agent $a \in \mathbb{A}\mathrm{gt}$; we assume that all $\mathfrak{Str}_a$ are disjoint.

**Definition 3.13 ($\mathcal{L}_{ATLI}$ [Jamroga et al., 2005]).** *Let* $\mathsf{p} \in \Pi$, $a \in \mathbb{A}\mathrm{gt}$, $A \subseteq \mathbb{A}\mathrm{gt}$, *and* $\sigma_a \in \mathfrak{Str}_a$. *The language* $\mathcal{L}_{ATLI}(\mathbb{A}\mathrm{gt}, \Pi, \mathfrak{Str})$ *is defined as:*

$$\theta ::= \mathsf{p} \mid \neg\theta \mid \theta \wedge \theta \mid \langle\langle A\rangle\rangle \bigcirc \theta \mid \langle\langle A\rangle\rangle \Box\theta \mid \langle\langle A\rangle\rangle \theta \mathcal{U} \theta \mid (\mathbf{str}_a \sigma_a)\theta.$$

Models for **ATLI** $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \mathcal{I}, \mathfrak{Str}, [\cdot]\rangle$ extend concurrent game structures with *intention relations* $\mathcal{I} \subseteq Q \times \mathbb{A}\mathrm{gt} \times Act$ (where $q\mathcal{I}_a\alpha$ means that $a$ possibly intends to do action $\alpha$ when in $q$). Moreover, strategic terms are interpreted as strategies according to function

$$[\cdot] : \mathfrak{Str} \to \bigcup_{a \in \mathbb{A}\mathrm{gt}} \Sigma_a^{ir} \text{ such that } [\sigma_a] \in \Sigma_a^{ir} \text{ for } \sigma_a \in \mathfrak{Str}_a$$

($\Sigma_a^{ir}$ denotes the set of $a$'s strategies). The set of paths consistent with all agents' intentions is defined as

$$\Lambda^{\mathcal{I}} = \{\lambda \in \Lambda_{\mathfrak{M}} \mid \forall i\ \exists \alpha \in d(\lambda[i])\ (o(\lambda[i], \alpha) = \lambda[i+1] \wedge \forall a \in \mathbb{A}\mathrm{gt}\ \lambda[i]\mathcal{I}_a\alpha_a)\}.$$

We impose on $\mathcal{I}$ the natural requirement that $q\mathcal{I}_a\alpha$ implies that $\alpha \in d_a(q)$ for $a \in \mathbb{A}\mathrm{gt}$; that is, agents only intend to do actions if they are actually able to perform them.

We say that strategy $s_A$ is *consistent with A's intentions* if $q\mathcal{I}_a s_A^a(q)$ for all $q \in Q, a \in A$. The *intention-consistent outcome set* is defined as:

$$out^{\mathcal{I}}(q, s_A) = out(q, s_A) \cap \Lambda^{\mathcal{I}}.$$

The semantics of strategic operators in **ATLI** extends and replaces the semantic rules of **ATL** as follows:

$\mathfrak{M}, q \models \langle\langle A\rangle\rangle \bigcirc \theta$ iff there is a collective strategy $s_A$ consistent with $A$'s intentions, such that for each $\lambda \in out^{\mathcal{I}}(q, s_A)$, we have that $\mathfrak{M}, \lambda[1] \models \theta$;
$\mathfrak{M}, q \models \langle\langle A\rangle\rangle\Box\theta$ and $\mathfrak{M}, q \models \langle\langle A\rangle\rangle\theta\mathcal{U}\theta'$: analogously;
$\mathfrak{M}, q \models (\mathbf{str}_a\sigma)\theta$ iff $revise(\mathfrak{M}, a, [\sigma]), q \models \theta$.

The function $revise(\mathfrak{M}, a, s_a)$ updates model $\mathfrak{M}$ by setting $a$'s intention relation to

$$\mathcal{I}_a = \{\langle q, s_a(q)\rangle \mid q \in Q\},$$

so that $s_a$ and $\mathcal{I}_a$ represent the same mapping in the resulting model. A pure CGS $\mathfrak{M}$ can be seen as a CGS with the full intention relation

$$\mathcal{I}^0 = \{\langle q, a, \alpha\rangle \mid q \in Q, a \in \mathbb{A}\mathrm{gt}, \alpha \in d_a(q)\}.$$

Additionally, for $A = \{a_{i_1}, \dots, a_{i_r}\}$ and $\sigma_A = \langle \sigma_{a_{i_1}}, \dots, \sigma_{a_{i_r}}\rangle$, we define: $(\mathbf{str}_A\sigma_A)\varphi \equiv (\mathbf{str}_{a_{i_1}}\sigma_{a_{i_1}})\dots(\mathbf{str}_{a_{i_r}}\sigma_{a_{i_r}})\varphi$. Furthermore, for $B = \{b_1, \dots, b_l\} \subseteq A$ we use $\sigma_A[B]$ to refer to $B$'s substrategy, i.e. to $\langle \sigma_{b_1}, \dots, \sigma_{b_l}\rangle$.

*Example 3.14 (Asymmetric matching pennies ctd.).* Coming back to our matching pennies example from Figure 3.5, we have for instance that $\mathfrak{M}_1, q_0 \models (\mathbf{str}_1\sigma)\langle\langle 2\rangle\rangle\Diamond\mathsf{money}_2$ if the denotation of $\sigma$ is set to $s_h$.

### 3.2.6 Further Logics for and about Games

Game Logic (**GL**)[Parikh, 1985] is another logic about games. It builds upon propositional dynamic logic (PDL) [Fischer and Ladner, 1979]. The logic can be used to reason about determined two player games [Blackburn et al., 2006]. The idea is to interpret the normal PDL operators in a game theoretic context and add some new constructs.

We would also like to mention the work [van Benthem, 2003] on rational dynamics and [Bonanno, 2002] on modal logic and game theory.

## 3.3 Logical Characterisation of Solution Concepts

In the following we show how game-theoretic solution concepts can be specified within **ATLI**. We discuss a classical approach and generalised solution concepts which are not based on utility values.

### 3.3.1 Standard Solution Concepts

Let $\sigma = \langle \sigma_1, \ldots, \sigma_k \rangle$ be a profile of strategic terms, and let $T$ stand for any of the following operators: $\bigcirc, \Box, \Diamond, \mathcal{U}\psi, \psi\mathcal{U}\_$ and let $a$ be an agent. Then we consider the following $\mathcal{L}_{ATLI}$ formulae:

$$BR_a^T(\sigma) \equiv (\mathbf{str}_{\mathbb{A}\mathrm{gt}\setminus\{a\}}\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]) \bigwedge_{v \in U} \Big( (\langle\!\langle a \rangle\!\rangle T \mathsf{p}_\mathsf{a}^\mathsf{v}) \rightarrow ((\mathbf{str}_a \sigma[a]) \langle\!\langle \emptyset \rangle\!\rangle T \mathsf{p}_\mathsf{a}^\mathsf{v}) \Big),$$

$$NE^T(\sigma) \equiv \bigwedge_{a \in \mathbb{A}\mathrm{gt}} BR_a^T(\sigma),$$

$$SPN^T(\sigma) \equiv \langle\!\langle \emptyset \rangle\!\rangle \Box NE^T(\sigma).$$

$BR_a^T(\sigma)$ refers to $\sigma[a]$ being a $T$-best strategy for $a$ against $\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]$; $NE^T(\sigma)$ expresses that strategy profile $\sigma$ is a T-Nash equilibrium; finally, $SPN^T(\sigma)$ defines $\sigma$ as subgame perfect T-NE. Thus, we have a family of equilibria: $\bigcirc$-Nash equilibrium, $\Box$-Nash equilibrium etc., each corresponding to a *different temporal pattern* of utilities. For example, we may assume that *agent a gets v* if a utility of at least $v$ is guaranteed for every time moment ($\Box \mathsf{p}_\mathsf{a}^\mathsf{v}$), is eventually achieved ($\Diamond \mathsf{p}_\mathsf{a}^\mathsf{v}$), and so on.

The correspondence between solution concepts and their temporal counterparts for extensive games is captured by the following proposition.

**Proposition 3.15.** *Let $\Gamma$ be an extensive game. Then the following holds:*

1. $\mathfrak{M}(\Gamma), \emptyset \models BR_a^{\Diamond}(\sigma)$ *iff* $[\sigma[a]]_{\mathfrak{M}(\Gamma)}$ *is a best response for a in* $\Gamma$ *against* $[\sigma[\mathbb{A}\mathrm{gt}\setminus\{a\}]]_{\mathfrak{M}(\Gamma)}$.

2. $\mathfrak{M}(\Gamma), \emptyset \models NE^{\Diamond}(\sigma)$ *iff* $[\sigma]_{\mathfrak{M}(\Gamma)}$ *is a NE in* $\Gamma$ *[Jamroga et al., 2005].*
3. $\mathfrak{M}(\Gamma), \emptyset \models SPN^{\Diamond}(\sigma)$ *iff* $[\sigma]_{\mathfrak{M}(\Gamma)}$ *is a SPN in* $\Gamma$.

*Proof.*

1. Note that $\mathfrak{M}(\Gamma)$ corresponds to an EF game so, the "payoff" propositions $\mathsf{p}_\mathsf{a}^\mathsf{v}$ can only become true at the "end" of each path in $\mathfrak{M}(\Gamma)$ (i.e. at the nodes that are reflexive and labelled by payoff propositions). Thus, $BR_a^{\Diamond}(\sigma)$ in $\mathfrak{M}(\Gamma), \emptyset$ holds iff, whenever $a$ can achieve the payoff of *at least* $v$ against $\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]$ (by any strategy for $a$), it can also achieve it by using $\sigma[a]$.
2. $s = (s_1, \ldots, s_k)$ is a NE iff $s_a$ is a best response to $s_{-a}$ for all $a$. Hence, $\mathfrak{M}(\Gamma), \emptyset \models NE^{\Diamond}(\sigma)$ iff for all $a$, $\mathfrak{M}(\Gamma), \emptyset \models BR_a^{\Diamond}(\sigma)$ iff (by 1) for all $a$, $[\sigma[a]]_{\mathfrak{M}(\Gamma)}$ is a best response for $a$ in $\Gamma$ against $[\sigma[\mathbb{A}\mathrm{gt}\setminus\{a\}]]_{\mathfrak{M}(\Gamma)}$ iff $[\sigma]_{\mathfrak{M}(\Gamma)}$ is a NE in $\Gamma$.
3. $\mathfrak{M}(\Gamma), \emptyset \models SPN^{\Diamond}(\sigma)$ iff $\mathfrak{M}(\Gamma), q \models NE^{\Diamond}(\sigma)$ for each $q$ reachable from the root $\emptyset$ (*). Since $\Gamma$ is a tree every node is reachable from $\emptyset$ in $\mathfrak{M}(\Gamma)$. So, by the second part, (*) iff $\sigma$ denotes a Nash equilibrium in every subtree of $\Gamma$.

■

We can use the above $\mathcal{L}_{ATLI}$-formulae to express game-theoretical properties of strategies in a straightforward way.

*Example 3.16 (Bargaining ctd.).* We extend the CGS in Figure 3.6 to a CGS with intentions; then, we have $\mathfrak{M}_2, q_0 \models NE^{\Diamond}(\sigma)$, with $\sigma$ interpreted in $\mathfrak{M}_2$ as $s^x$ (for any $x \in [0, 1]$). Still, $\mathfrak{M}_2, q_0 \models SPN^{\Diamond}(\sigma)$ if, and only if, $[\sigma]_{M_2} = s^\kappa$.

We also propose a (tentative) $\mathcal{L}_{ATLI}$ characterisation of *Pareto optimality.* For normal form games we have the following characterisation [van der Hoek et al., 2005a] :

$$PO^T(\sigma) \equiv \bigwedge_{v_1} \cdots \bigwedge_{v_k} \Big( (\langle\!\langle \mathbb{A}\mathrm{gt} \rangle\!\rangle T \bigwedge_i \mathsf{p}_\mathsf{i}^{\mathsf{v_i}}) \rightarrow$$
$$(\mathbf{str}_{\mathbb{A}\mathrm{gt}}\sigma)\big((\langle\!\langle \emptyset \rangle\!\rangle T \bigwedge_i \mathsf{p}_\mathsf{i}^{\mathsf{v_i}}) \vee (\bigvee_i \bigvee_{\substack{v' \text{ s.t.} \\ v' > v_i}} \langle\!\langle \emptyset \rangle\!\rangle T \mathsf{p}_\mathsf{i}^{\mathsf{v}'})\big)\Big).$$

That is, the strategy profile denoted by $\sigma$ is Pareto optimal iff, for every achievable pattern of payoff profiles, either it can be achieved by $\sigma$, or $\sigma$ obtains a strictly better payoff profile for at least one player. We note that the above formula has exponential length with respect to the number of payoffs in $U$. Moreover, it is not obvious that this characterisation is the intuitively right one, as it refers in fact to the evolution of payoff *profiles* (i.e., combinations of payoffs achieved by agents at the same time), and not on temporal patterns of payoff evolutions for *each* agent separately. So, for example, $PO^{\Diamond}(\sigma)$ may hold even if there is a strategy profile $\sigma'$ that makes each agent achieve eventually a better payoff, as long as not all of them will achieve these better payoffs at the same moment. Still, the following holds.

**Proposition 3.17.** *Let $\Gamma$ be an extensive game. Then $\mathfrak{M}(\Gamma), \emptyset \models PO^{\Diamond}(\sigma)$ iff $[\sigma]_{M(\Gamma)}$ is Pareto optimal in $\Gamma$.*

*Proof.* "$\Rightarrow$": Let $\mathfrak{M}(\Gamma), \emptyset \models PO^{\Diamond}(\sigma)$. Then, for each payoff profile $\langle v_1, \ldots, v_k \rangle$ reachable in $\Gamma$, we have that either $[\sigma]$ obtains an at least as good profile,[3] or it obtains an incomparable payoff profile. Thus, $[\sigma]$ is Pareto optimal. "$\Leftarrow$":The proof for the other direction is done analogously. ■

*Example 3.18 (Asymmetric matching pennies ctd.).* Let $\mathfrak{M}_1'$ be our matching pennies model $\mathfrak{M}_1$ with additional propositions $\mathsf{p}_\mathsf{i}^1 \equiv \mathsf{money_i}$ (so, we assign to $\mathsf{money_i}$ a utility of 1 for $i$). Then, we have $\mathfrak{M}_1', q_0 \models PO^{\Diamond}(\sigma)$ iff $\sigma$ denotes the strategy profile $\langle s_h, s_h \rangle$.

### 3.3.2 General Solution Concepts

We also propose an alternative approach to defining solution concepts for games that involve an infinite flow of time. In the new approach, path formulae of $\mathcal{L}_{ATL}$ are used to specify the "winning conditions" of each player. This implicitly leads to a normal form game, where the traditional solution concepts are well defined. We also demonstrate how these "qualitative" solution concepts (parametrised by $\mathcal{L}_{ATL}$-path formulae) can be characterised in **ATLP**. In the following we sketch the idea of *general solution concepts*. We shall elaborate on these concepts in Section 6.4.2, using the logic **ATLP**.

We have seen in Section 3.2.2 that some (but not all!) concurrent game structures can be seen as extensive form games. These CGSs are be turn-based (i.e., players play by taking turns) and have a tree-like structure; moreover, they include special propositions that emulate payoffs and can be used to define agents' preferences. Now, we want to extend the correspondence to arbitrary CGSs. Our idea is to *determine the outcome of a game by the truth of certain path formulae* (e.g., in the case of binary payoffs, we can see the formulae as *winning conditions*). So, we give up the idea of assigning payoffs to leaves in a tree. Instead, we see a concurrent game structure as a game, paths in the structure as plays in the game, and satisfaction of some pre-specified formula as the mechanism that defines agents' outcome for a given play.

Which formulae can be used in this respect? We propose that player $i$'s preferences can be specified by a finite list of $\mathcal{L}_{ATL}$-path formulae $\eta_i =$

[3] We recall that $\bigwedge_i \mathsf{p}_\mathsf{i}^{\mathsf{v_i}}$ means that each player $i$ gets *at least* $v_i$.

$\langle \eta_i^1, \ldots, \eta_i^{n_i} \rangle$ (where $n_i \in \mathbb{N}$) with the underlying assumption that agent $i$ prefers $\eta_i^1$ most, $\eta_i^2$ comes second etc., and the worst outcome occurs when no $\eta_i^1, \ldots, \eta_i^{n_i}$ holds for the actual play. Thus, $\eta_i$ imposes a total order on paths in a CGS.

For $k$ players, we need a $k$-vector of such preference lists $\overrightarrow{\eta} = \langle \eta_1, \ldots, \eta_k \rangle$. Then, every concurrent game structure gives rise to the strategic game defined as below.

**Definition 3.19 (From CGS to NF game).** *Let $\mathfrak{M}$ be a* CGS, $q \in Q_{\mathfrak{M}}$ *a state, and $\overrightarrow{\eta} = \langle \eta_1, \ldots, \eta_k \rangle$ a vector of lists of $\mathcal{L}_{ATL}$-path formulae, where $k = |\mathbb{A}\mathrm{gt}|$.*

*Then we define $\mathcal{S}(\mathfrak{M}, \overrightarrow{\eta}, q)$, the* NF game *associated with $\mathfrak{M}$, $\overrightarrow{\eta}$, and $q$, as the strategic game $\langle \mathbb{A}\mathrm{gt}, \mathcal{A}_1, \ldots, \mathcal{A}_k, \mu \rangle$, where the set $\mathcal{A}_i$ of $i$'s strategies is given by $\Sigma_i$ for each $i \in \mathbb{A}\mathrm{gt}$, and the payoff function is defined as follows:*

$$\mu_i(a_1, \ldots, a_k) = \begin{cases} n_i - j + 1 & \text{if } \eta_i^j \text{ is the first formula from } \eta_i \text{ such that} \\ & \mathfrak{M}, \lambda \models \eta_i^j \text{ for all } \lambda \in out(q, \langle a_1, \ldots, a_k \rangle), \\ 0 & \text{no } \eta_i^j \text{ is satisfied} \end{cases}$$

*where $\eta_i = \langle \eta_i^1, \ldots, \eta_i^{n_i} \rangle$, $1 \leq j \leq n_i$ and we write $\mu_i$ for $\mu(i)$.*

Below, we present the generalised version of temporal Nash equilibrium and temporal subgame perfect NE:

$$\begin{aligned} BR_a^{\overrightarrow{\eta}}(\sigma) &\equiv (\mathbf{str}_{\mathbb{A}\mathrm{gt}\setminus\{a\}}\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]) \\ &\qquad \bigwedge_{j \in \{1,\ldots n_a\}} \Big( (\langle\!\langle a \rangle\!\rangle \eta_a^j) \to ((\mathbf{str}_a \sigma[a]) \bigvee_{r \leq j} \langle\!\langle \emptyset \rangle\!\rangle \eta_a^r) \Big), \\ NE^{\overrightarrow{\eta}}(\sigma) &\equiv \bigwedge_{a \in \mathbb{A}\mathrm{gt}} BR_a^{\overrightarrow{\eta}}(\sigma), \\ SPN^{\overrightarrow{\eta}}(\sigma) &\equiv \langle\!\langle \emptyset \rangle\!\rangle \Box NE^{\overrightarrow{\eta}}(\sigma). \end{aligned}$$

In Section 6.4.2 we do also provide proofs that these characterisations (in terms of the logic **ATLP**) correspond to their game-theoretic counterparts.

The case with a single "winning condition" per agent is particularly interesting. Clearly, it gives rise to a normal form game with binary payoffs (cf., for instance, our informal discussion of the "matching pennies" variant in Example 3.10). We will stick to such binary games throughout the rest of this thesis (especially in Section 6.4.2 where general solution concepts are studied in more detail), but one can easily imagine how the binary case extends to the case with multiple levels of preference.

# 4

# Background on Various Things Needed

## Contents



Throughout this thesis we make use of different sorts of theories. In this chapter we introduce the basic concepts needed. Firstly, we consider complexity theory. Then, we proceed with (abstract) argumentation from a coalition formation perspective. Finally, we turn to probability theory and Petri nets.

## 4.1 Background in Complexity Theory

Complexity theory plays an important role in many applications of computer science. We will make use of it when determining the model checking complexity. We begin with introducing *Turing machines* (TMs), present the relevant complexity classes, and recall basic results about them. Then, we introduce the notions of reduction and completeness and present some complete problems which we are going to use in the remainder of the thesis. For a more detailed introduction we refer the reader to [Papadimitriou, 1994; Hopcroft and Ullman, 1979].

### 4.1.1 Turing Machines

In this section we recall the basic notions of computation. The presentation follows mostly [Papadimitriou, 1994].

A *Turing machine* (TM) has an infinite read/write *band*. The machine can move its read/write head to the left, to the right, or write some symbol drawn from a *finite alphabet* $\Sigma$, respectively. We assume the presence of a special symbol # (the *blank* symbol) in $\Sigma$. This symbol is not permitted as an input symbol. The mode of operation depends on the symbol read and the current state of the machine. The set of *states* $K$ is finite. We assume that the initial state $s^I$ belongs to $K$. A TM *halts* on an input if it reaches the halting state $h \notin K$ from the initial state $s^I$. Two additional halting states $Y, N \notin K$ indicate that the machine accepts (Y) and rejects (N) the input, respectively. Formally, a TM is given by a tuple

$$\mathcal{A} = (K, \Sigma, \delta)$$

where $\delta$ is the *transition function.* If $\mathcal{A}$ is a *deterministic* TM the function is given by

$$\delta : K \times \Sigma \rightarrow K \cup \{h, Y, N\} \times (\Sigma \cup \{L, R\}).$$

In case that the TM is *non-deterministic* we define

$$\delta : K \times \Sigma \rightarrow \mathcal{P}(K \cup \{h, Y, N\} \times (\Sigma \cup \{L, R\})).$$

The transition $(q', L) = \delta(q, a)$ (resp. $(q', L) \in \delta(q, a)$) indicates that the machine changes its state to $q'$ and moves its head to the left provided that the current state is $q$ and symbol $a$ is under the read/write head.

A *configuration* of a TM is a word from $K \cup \{h, Y, N\} \times \Sigma^+ \times \Sigma^*$. A configuration $(q, vx, y)$, $x \in \Sigma$, encodes that the current state is $q$, the cell on which the head is contains symbol $x$ (the machine *reads* $x$); the word left of the head is given by the (possibly empty) word $v$ and the word right of the head by the (possibly empty) word $y$. We assume that $v$ and $y$ contain only necessary information; that is, no unnecessary blank symbols on the left (resp. right) of the last non-blank symbol left (resp. right) from $x$.

A *computation* of a TM is a sequence of configurations that can result from the operation of the machine. The state of the initial configuration is given by $s^I$. We say that a computation is *halting* if it is finite and the state of the final configuration is among $h, Y, N$. If there is a halting configuration we say that the machine *halts*; in notation, $\mathcal{A}(w) \downarrow$: The Turing machine $\mathcal{A}$ halts on input $w$. Otherwise, we say the machine does not halt and we write $\mathcal{A}(w) \not\downarrow$.

For deterministic TMs we also define what it means to *decide a language*. A configuration is said to be *accepting* (resp. *rejecting*) if it is halting and the state of the final configuration is $Y$ (resp. $N$). We write $\mathcal{A}(w) \downarrow_Y$ and $\mathcal{A}(w) \downarrow_N$, respectively. We say that a deterministic TM $\mathcal{A}$ *accepts* a language $L$ if $\mathcal{A}(w) \downarrow_Y$ (resp. $\mathcal{A}(w) \not\downarrow$) for all $w \in L$ (resp. $w \notin L$). Similarly, a $\mathcal{A}$ is said to *decide* a language $L$ if $\mathcal{A}(w) \downarrow_Y$ (resp. $\mathcal{A}(w) \downarrow_N$) for all $w \in L$ (resp. $w \notin L$). Finally, a language is said to be *acceptable* (resp. *decidable*) if there is a deterministic TM that accepts (resp. decides) the language. Note, that the halting state can be used to *compute* functions. Given an input $x$ the output, denoted by $\mathcal{A}(w)$, is the content on the band once the machine has halted, provided that $\mathcal{A}(w) \downarrow$.

In order to define the complexity of languages/problems it is important how many steps a machine does and how many space cells it uses. We say that a machine $\mathcal{A}$ needs time (resp. space) $k$ to accept input $w$ if it makes $k$ steps (resp. needs $k$ storage cells) to accept $w$, provided it accepts $w$. The "best" (i.e. shortest) accepting configuration is considered. Now, let $f : \mathbb{N} \to \mathbb{N}$ be some function. We say that machine $\mathcal{A}$ is $f$-time bounded (resp. $f$-space bounded) if it accepts/decides every accepted input $w$ of size $n$ within $f(n)$ steps (resp. $f(n)$ space cells). In the case of space, one considers *offline* TMs. These machines have a read-only input tape and a separate readable and writeable working tape. Only the latter counts in the calculation in terms of space. For further details we refer to [Papadimitriou, 1994].

Finally, we define **DTIME**$(f)$ (resp. **DSPACE**$(f)$) as the set of all languages acceptable by some $f$-time bounded (resp. $f$-space bounded offline) deterministic TM. The classes **NTIME**$(f)$ and **NSPACE**$(f)$ are analogously defined for non-deterministic machines.

### 4.1.2 Standard Complexity Classes between P and PSPACE

We have provided the very basic concept of TMs. In the following we consider standard complexity classes and recall some important results.

In general, a *complexity class* is a set of languages that share some computational properties regarding their acceptance by TMs. The most basic classes are the set of languages acceptable by deterministic and non-deterministic Turing machines in *polynomial time* and *polynomial space*. These classes are called **P** (polynomial deterministic time), **NP** (polynomial non-deterministic time), and **PSPACE** (polynomial deterministic space), respectively. We did not introduce the class "polynomial non-deterministic space" as this class coincides with **PSPACE** by Savitch's well-known result. The following relation is obvious:

$$\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE}.$$

Given a complexity class $\mathcal{C}$ we use $\mathbf{co}\mathcal{C}$ (complementary class of $\mathcal{C}$) to denote the class of languages with their complement being in $\mathcal{C}$.

The finer-grained classification of the space between $\mathbf{P}$ and $\mathbf{PSPACE}$ is described by the *polynomial hierarchy* (cf. [Papadimitriou, 1994; Garey and Johnson, 1979]). The classes are recursively defined as follows.

**Definition 4.1 (Polynomial hierarchy).** *We define the classes* $\mathbf{\Delta_0^P} = \mathbf{\Sigma_0^P} = \mathbf{\Pi_0^P} := \mathbf{P}$ *and for* $i \geq 0$

$$\mathbf{\Delta_{i+1}^P} := \mathbf{P}^{\mathbf{\Sigma_i^P}}, \qquad \mathbf{\Sigma_{i+1}^P} := \mathbf{NP}^{\mathbf{\Sigma_i^P}}, \qquad \mathbf{\Pi_{i+1}^P} := \mathbf{coNP}^{\mathbf{\Sigma_i^P}}$$

*and the complete* polynomial hierarchy *as* $\mathbf{PH} := \bigcup_{i \geq 0} \mathbf{\Sigma_i^P}$.

We have the following relation:

$$\mathbf{P} = \mathbf{\Delta_1^P} \subseteq \mathbf{NP} = \mathbf{\Sigma_1^P} \subseteq \mathbf{\Delta_2^P} \subseteq \mathbf{\Sigma_2^P} \subseteq \mathbf{\Delta_3^P} \subseteq \cdots \subseteq \mathbf{PH} \subseteq \mathbf{PSPACE}.$$

The following two results are useful to prove that languages reside in some class of the polynomial hierarchy. Firstly, we need two more notations. A relation $R \subseteq \Sigma^* \times \Sigma^*$ is said to be *polynomially time decidable* if the language $\{(x, y) \mid (x, y) \in R\}$ is in $\mathbf{P}$. Such a relation is said to be *polynomially balanced* if there is a number $k$ such that if $(x, y) \in R$ implies that $|y| \leq |x|^k$.

**Theorem 4.2 ([Papadimitriou, 1994]).** *Let $L$ be a language and $i \geq 1$. Then, we have that $L \in \mathbf{\Sigma_i^P}$ if, and only if, there is a polynomially balanced relation $R$ such that $\{x; y \mid (x, y) \in R\} \in \mathbf{\Pi_{i-1}^P}$ and $L = \{x \mid \exists y\ ((x, y) \in R)\}$.*

**Corollary 4.3 ([Papadimitriou, 1994]).** *For any language $L$ and $i \geq 0$ we have the following. $L \in \mathbf{\Sigma_i^P}$ if, and only if, there is a polynomially balanced, polynomially time decidable $(i + 1)$-ary relation $R$ such that*

$$L = \{x \mid \exists y_1 \forall y_2 \ldots Q y_i\ ((x, y_1, \ldots, y_i) \in R)\}$$

*where $Q = \forall$ (resp. $Q = \exists$) if $i$ is even (resp. odd).*

### 4.1.3 The Complexity Classes PP, #P, and $\mathbf{P^{\#P}}$

In this section we consider some more exotic complexity classes which will be used in the thesis.

An input is accepted by a non-deterministic TM if at least one of the machine's computations is accepting. It is possible that all other computations are indeed rejecting. The complexity class $\mathbf{PP}$ (probabilistic polynomial time) is based on the idea that at least half of the computations must be accepting in order to accept the input.

**Definition 4.4 (PP [Papadimitriou, 1994; Beigel et al., 1995]).** *Let $\mathcal{A}$ be a non-deterministic* TM *and $w$ an input. By $Pos(\mathcal{A}, w)$ we denote the number of accepting computations minus the number of non-halting computations of $\mathcal{A}$ on $w$.*

*The complexity class* **PP** *contains all languages $L$ for which there is a polynomial time bounded non-deterministic* TM *such that for all $w \in \Sigma^*$ it holds that*

$$Pos(\mathcal{A}, w) > 0 \textit{ if, and only if, } w \in L.$$

Alternatively, one may define **PP** according to its original definition in terms of *probabilistic* TMs [Gill, 1977].

It is easily seen, that a language in **NP** is also in **PP**. One can simply add accepting computations such that just one more (the accepting one of the **NP** problem) is needed for a majority of accepting computations. Intuitively, the new machine nondeterministically guesses to work in the very same way as the old machine or to work like a slightly modified version of it. The difference of the modified version is that every computation is accepting.

**Theorem 4.5 ([Papadimitriou, 1994]). NP $\subseteq$ PP**

In [Beigel et al., 1995] it is shown that the class **PP** is closed under various operations including intersection and union. In addition to that in [Gill, 1977] the closure under complementation is shown.

**Theorem 4.6 ([Gill, 1977; Beigel et al., 1995]).** *The class* **PP** *is closed under union, intersection, and complement.*

From that we do also get that

$$\mathbf{coNP} \subseteq \mathbf{PP}.$$

The real power of the class **PP** is seen if it is used as an oracle of a polynomial time deterministic TM; then, it contains the whole polynomial hierarchy.

**Theorem 4.7 ([Toda, 1989]). PH $\subseteq$ $\mathbf{P}^{\mathbf{PP}}$**

A different type of complexity class is #**P**. Although this class contains functions (instead of languages) it is closely related to **PP**. Suppose $R$ is a polynomially balanced, polynomial time decidable binary relation. Then, we define the *R-counting problem* as the function that returns the number of strings $y$ such that $(x, y) \in R$ given $x$ as input. So, the solution to the $R$-counting problem can be seen as a function $f_R : \Sigma^* \to \mathbb{N}_0$; $f_R(x)$ denotes the number of $y$'s such that $(x, y) \in R$. The complexity class #**P** contains all such functions that are associated with an $R$-counting problem [Papadimitriou, 1994].

**Definition 4.8 (#P[Papadimitriou, 1994]).** *The complexity class* **#P** *contains the function $f_R$ associated with the R-counting problem for each polynomially balanced, polynomial time decidable binary relation R.*

The relation to the probabilistic class **PP** is shown by the following interesting result.

**Theorem 4.9 ([Angluin, 1980]).** $\mathbf{P}^{\mathbf{PP}} = \mathbf{P}^{\#\mathbf{P}}$

### 4.1.4 Reductions, Completeness, and Decidability

In order to show the difficulty of our model checking problems we use them to solve other problems already shown to be difficult in one sense or another. Formally, this is captured by *reductions*. Let $L_1$ and $L_2$ be two languages. A *reduction* $R$ is a function that transforms each instance $w$ of $L_1$ to an instance $R(w)$ of $L_2$ such that

$$w \in L_1 \text{ if, and only if, } R(w) \in L_2.$$

This shows that the problem $L_2$ is at least as hard as $L_1$ provided that the calculation of the reduction itself is not too difficult. We say that $L_1$ is *reducible* to $L_2$.

**Definition 4.10 (Reduction, complexity).** *A language $L_1$ is* $f$-space (resp. $f$-time) reducible *to $L_2$ if there is a function $R : \Sigma^* \to \Sigma^*$ computable by a $f$-space (resp. $f$-time) bounded deterministic* TM *such that*

$$w \in L_1 \textit{ if, and only if, } R(w) \in L_2.$$

*Then, we write $L_1 \leq_f L_2$ and call $R$ a reduction of $L_1$ to $L_2$. In this paper, we will usually consider logarithmic space or polynomial time reductions.*

It is important to note that the following definition does strongly depend on the specific reduction used. For example, if one considers the complexity class **P** and polynomial time reductions any problem from **P** is trivially **P**-complete. Reductions have to be chosen carefully.

**Definition 4.11 (Hardness and Completeness).** *Given a complexity class $\mathcal{C}$ and a language $L$ we say that $L$ is* $\mathcal{C}$-hard *under $f$-time (resp. $f$-space) reductions if any problem from $\mathcal{C}$ can be $f$-time (resp. $f$-space) reduced to $L$. $L$ is called* $\mathcal{C}$-complete *if additionally $L \in \mathcal{C}$.*

Reductions can also be used to show that problems are undecidable. Instead of classifying reductions according to their computational complexity one can simply consider decidable reductions. Then, if one reduces an undecidable problem $L_1$ to a problem $L_2$ the latter must also be undecidable. Similarly, if a problem $L_1$ is reduced to a decidable problem $L_2$; then, $L_1$ must also be decidable.

**Definition 4.12 (Reduction, Decidability).** *A language $L_1$ is* reducible *to $L_2$ if there is a total decidable function $R : \Sigma^* \to \Sigma^*$ such that*

$$w \in L_1 \text{ if, and only if, } R(w) \in L_2.$$

All these reductions are called *many-to-one* reductions. A second kind of (weaker) reductions are called *Turing reductions.* A problem $L_1$ is *Turing-reducible* to $L_2$ iff $L_1$ can be decided by help of an $L_2$-oracle TM.

### 4.1.5 Some Complete Problems

To show that a problem is *hard* with respect to some complexity class (i.e. that there is no other problem in the complexity class of interest that is significantly harder to solve) one often reduces problems already known to have this properties to the new one. Here, we introduce some of these known problems that we are going to facilitate in this thesis.

A typical **PSPACE**-complete problem is *quantified satisfiability* (QSAT), given a quantified Boolean formula one is interested whether it is satisfiable.

**Definition 4.13** (QSAT **[Papadimitriou, 1994; Garey and Johnson, 1979]).**
***Input:*** *A Boolean formula $\varphi$ with $i$ variables $x_1, \ldots, x_i$.*
***Output:*** *True if $\exists x_1 \forall x_2 \ldots Q_i x_i\ \varphi$ is satisfiable, false otherwise (where $Q = \forall$ if $i$ is even, and $Q = \exists$ if $i$ is odd).*

**Theorem 4.14 ([Papadimitriou, 1994; Garey and Johnson, 1979]).** QSAT *is* **PSPACE**-*complete.*

The input formula is often assumed to be in a more restricted form without changing the problem's complexity. One might assume that the input is presented in conjunctive normal form (CNF), or even in CNF with only 3 literals per clause [Garey and Johnson, 1979], or in negation normal form (NNF) (that is, negations occur only at literals). Simple rewrite rules allow to obtain the latter normal form.

The number of alternations in QSAT is unbounded. The problem which contains all such formulas up to a bounded number of alternations, say $k$, is shown to be $\mathbf{\Sigma_k^P}$-complete in [Meyer and Stockmeyer, 1972]. We are interested in an apparently harder variant of these problems.

**Definition 4.15** (SNSAT$_i$,**[Laroussinie et al., 2008]).**
***Input:*** *$p$ sets of propositional variables $X_r^j = \{x_{1,r}^j, ..., x_{k,r}^j\}$ for each $j = 1, \ldots, i$; $p$ propositional variables $z_r$, and $p$ Boolean formulae $\varphi_r$ in positive normal form (i.e., negation is allowed only on the level of literals) for $r = 1, \ldots, p$. Each $\varphi_r$ involves only variables from $\bigcup_{j=1}^{i} X_r^j \cup \{z_1, ..., z_{r-1}\}$, with the following requirement:*

$z_r \equiv \exists X_r^1 \forall X_r^2 \exists X_r^3 \dots Q X_r^i . \varphi_r(z_1, ..., z_{r-1}, X_r^1, \dots, X_r^i)$ *where* $Q = \forall$ *(resp.* $Q = \exists$*) if* $i$ *is even (resp. odd).*
***Output:*** *The value of* $z_p$*.*

**Theorem 4.16 ([Laroussinie et al., 2008]).** $\mathrm{SNSAT}_i$ *is* $\mathbf{\Delta_i^P}$*-complete for* $i \geq 1$*.*

The following problem is the "satisfiability problem" for **PP**. A Boolean formula belongs to MAJSAT if more than half of the truth assignments satisfy the formula.

**Definition 4.17** (MAJSAT)**.** *Given a formula* $\varphi$ *in* CNF *with propositional variables* $x_1, \dots, x_n$*, answer YES if more than half of all assignments of* $x_1, \dots, x_n$ *make* $\varphi$ *true, and NO otherwise.*

**Theorem 4.18 ([Papadimitriou, 1994]).** MAJSAT *is* **PP***-complete.*

In the following, we consider some graph theoretical problem.

**Definition 4.19 (Graph reachability).** *Let* $G = (V, E)$ *be graph. Given two vertices* $u, v \in V$ *the* graph-reachability problem *is the question whether* $v$ *is reachable from* $u$*.*

**Theorem 4.20 ([Jones, 1977, 1975]).** *The graph-reachability problem is* **NLOGSPACE***-complete under logarithmic space reductions.*

An *and-or graph* (or *alternating graph*)[Immerman, 1981] is a tuple $(G, l)$ such that $G = (E, V)$ is a directed acyclic graph and $l : V \to \{\wedge, \vee\}$ a function labelling each state of $G$ either as an or-node ($\vee$) or as an and-node ($\wedge$). Given two vertices $u$ and $v$ of $G$ we define what it means that $v$ is *reachable* from $u$. Let $x_1, \dots, x_n$ denote all successor nodes of $u$. Then, $v$ is said to be *reachable* from $u$ iff

1. $u = v$; or
2. $l(u) = \wedge$, $n \geq 1$, and $v$ is reachable from all $x_i$'s; or
3. $l(u) = \vee$, $n \geq 1$, and $v$ is reachable from some $x_i$.

**Definition 4.21 (And-Or-Graph Reachability Problem [Immerman, 1981]).** *Let an and-or graph* $((V, E), l)$ *and two vertices* $u, v \in V$ *be given. The* and-or-graph reachability problem *is the question whether* $v$ *is reachable from* $u$*.*

**Theorem 4.22 ([Immerman, 1981]).** *The and-or-graph reachability problem is* **P***-complete.*

The following problem is used to establish the lower bound of **ATL**$^*$ model checking. Let $\lambda$ be a path; i.e. an $\omega$-sequence of states. An $x$-labelling is a function $\pi^x : \mathbb{N}_0 \to \{x, \emptyset\}$. It can be used to label position $\lambda[i]$ of path $\lambda$ with $\pi^x(i)$; hence, such a function may label some positions of the path with $x$. An $(x, y)$-*program* is a function $P^{xy} : \{x, \emptyset\}^+ \to \{y\}$. Such an $(x, y)$-program together with a $\pi^x$-labelling induces a $y$-labelling $\pi^y$ as follows:

$$\pi^y(i) = y \text{ iff } P^{xy}(\pi^x(\lambda[0]) \dots \pi^x(\lambda[i])) = y.$$

Hence, a $\pi^y$ labelling labels states of $\lambda$ with a proposition $y$ subject to the labels assigned to the path by a $\pi^x$-labelling following a given $(x, y)$-program $P^{xy}$. Given a $\pi^x$ and $\pi^y$ labelling we use $\pi^{x,y}$ to denote the combined labelling

$$\pi^{x,y}(\cdot) = \pi^x(\cdot) \cup \pi^y(\cdot).$$

**Definition 4.23 (LTL-realisability [Pnueli and Rosner, 1989; Rosner, 1992]).** *Let $\lambda$ be a path. An $\mathcal{L}_{LTL}(\{x, y\})$-formula $\varphi(x, y)$ over propositions $x$ and $y$ is said to be* realisable *iff for any $x$-labelling $\pi^x$ there is an $(x, y)$-program such that for the induced $\pi^y$ labelling we have that $\lambda, \pi^{x,y} \models^{\mathsf{LTL}} \varphi(x, y)$.*

**Theorem 4.24 ([Pnueli and Rosner, 1989; Rosner, 1992]).**
**LTL**-*realisability is* **2EXPTIME**-*complete.*

### 4.1.6 Automata Theory

In Section 5 we often present automata-based model checking algoritms. We assume that the reader is familiar with *finite automata on finite words*. In this section we briefly introduce *finite automata on infinite words and trees*.

#### Automata on Infinite Words

**Definition 4.25 ($\omega$-automaton).** *An* $\omega$-automaton *is a quintuple*

$$A = (Q, \Sigma, \Delta, q_I, C)$$

*where*

- *$Q$ is a finite set of states;*
- *$\Sigma$ is a finite alphabet;*
- *$\Delta \subseteq Q \times \Sigma \times Q$ a transition relation; and*
- *$C$ an acceptance component (which is specified in the following).*

**Definition 4.26 (Run).** *A* run *of $A$ $\rho$ on a word $w = w_1 w_2 \dots \in \Sigma^\omega$ is an infinite sequence of states of $A$ $\rho = \rho(0)\rho(1) \dots \in Q^\omega$ such that:*

*1.* $\rho(0) = q_I$ *and*
*2.* $\Delta(\rho(i-1), w_i, \rho(i)) \in \Delta$ *for* $i \geq 1$.

We define $\mathit{Inf}(\rho)$ as the set of *all states that occur infinitely often* on $\rho$; that is,

$$\mathit{Inf}(\rho) = \{q \in Q \mid \forall i \exists j (j > i \wedge \rho(j) = q\}.$$

Depending on the accepting condition various types of automata arise.

**Definition 4.27 (Büchi automaton).** *A* Büchi automaton *is an $\omega$-automaton*

$$A = (Q, \Sigma, \Delta, q_I, F)$$

*where $F \subseteq Q$ with the following* acceptance condition*: $A$* accepts *$w \in \Sigma^\omega$ if, and only if, there is a run $\rho$ of $A$ on $w$ such that*

$$\mathit{Inf}(\rho) \cap F \neq \emptyset.$$

Thus, such an automaton accepts all words such that some state from $F$ is visited infinitely often on a corresponding run.

**Definition 4.28 (Language).** *The* language *of $A$, $L(A)$ consists of all words accepted by $A$; that is,*

$$L(A) = \{w \in \Sigma^\omega \mid A \text{ accepts } w\}.$$

**Theorem 4.29 (Characterisation of $\omega$-languages).** *A language $L$ is* Büchi acceptable *if, and only if, there are finitely many regular languages $U_1, \ldots, U_n$ and $V_1, \ldots, V_n$ such that*

$$L = \bigcup_{i=1,\ldots,n} U_i (V_i)^\omega$$

**Corollary 4.30 ([Vardi and Wolper, 1994]).** *Any Büchi recognisable non-empty language $L$ contains an* ultimately periodic word*.*

For the model checking algorithms we need to check whether the language of a Büchi automaton is empty or not.

**Theorem 4.31 ([Vardi and Wolper, 1994]).** *The* non-emptiness problem for Büchi automata **NLOGSPACE***-complete under logarithmic space reductions.*

*Proof.* [Sketch] We check if there is some ultimately periodic word by determining a reachable accepting state that is reachable from itself. The following algorithm runs in non-deterministic logarithmic space: 1) Guess an initial state $i$ and accepting state $r$. 2) Check whether $\mathit{reach}(i, r)$ and $\mathit{reach}(r, r)$ where *reach*$(x, y)$ choses a transition from $x$ to some successor $x'$ and returns "yes" if $x' = y$; otherwise, recursively performs *reach*$(x', y)$.

Hardness is shown by a reduction of the **NLOGSPACE**-complete problem Graph reachability from Definition 4.19. The reduction is straightforward. Given $G, u, v$, transform $G$ to a Büchi automaton with initial state $u$ and final state $v$ and add a loop to $v$. ■

## Automata on Infinite Trees

As we will later see, $\omega$-automata provide means to model check linear-time logics. For branching time or strategic logics *tree automata* can be used. Such automata do not focus on $\omega$-ordered linear sequences but rather on infinite trees.

As before let $\Sigma$ be a finite alphabet and $k$ a natural number. A *$k$-ary $\Sigma$-tree* $t = (dom_t, L)$ is a tree with maximal branching $k$ and in which each node is labelled by an element from $\Sigma$. That is

$$L : dom_t \to \Sigma$$

where $dom_t \subseteq \{0, \ldots, k-1\}^*$ denotes the *domain* of the tree. It is required that $dom_t$ is closed under prefixes, i.e.

$$wx \in dom_t \to \forall y (0 \leq y < x \to wy \in dom_t).$$

A $k$-ary $\omega$-tree automaton over the alphabet $\Sigma$ is an automaton that accepts infinite $k$-ary $\Sigma$-trees.

**Definition 4.32 ($k$-ary $\omega$-tree automaton).** *A* $k$-ary $\omega$-tree automaton *over the alphabet $\Sigma$ is given by a tuple*

$$A = (Q, q_I, \Delta, C)$$

*where*

- *$Q$ is a set of states,*
- *$q_I \in Q$ the initial state,*
- *$\Delta : Q \times \Sigma \times \{1, \ldots, k\} \to \mathcal{P}(\cup_{i=1\ldots k} Q^i)$ with $\Delta(q, a, i) \subseteq Q^i$ a transition relation, and*
- *$C$ an acceptance component (which is specified in the following).*

**Definition 4.33 (Run, path, successful, accepting).** *A* run *of a $k$-ary $\omega$-tree automaton $A$ on an infinite $k$-ary $\Sigma$-tree $t = (dom_t, L_t)$ is an infinite $k$-ary $Q$-tree $r = (dom_r, L_r)$ such that*

*1. $dom_r = dom_t$,*

*2.* $L_r(\emptyset) = q_I$ *and*
*3.* $\forall w \in dom_t : (L_r(w0), \ldots, L_r(wi)) \in \Delta(L_r(w), L_t(w), i)$ *where* $i = \max\{j \mid wj \in dom_t\}$.

*A* path *of the run* $r$ *is an infinite linearly ordered subset of* $dom_r$ *(i.e. it denotes a branch in the tree). We say that run* $r$ *is* successful *if each path of* $r$ *satisfies the accepting condition* $C$*. An input tree* $t$ *is* accepted *by* $A$ *if there is a successful run.*

Finally, we instantiate the acceptance condition of $\omega$-tree automata and obtain Büchi and Rabin tree automata.

**Definition 4.34 (Büchi tree automaton).** *A Büchi tree automaton is given by an* $\omega$*-tree automaton* $A = (Q, q_I, \Delta, F)$ *where* $F \subseteq Q$ *is a set of final states. A run* $r = (dom_r, L)$ *is* successful *if, and only if, for each path* $p$ *on* $r$ *there is a state that occurs infinitely often on* $p$*; i.e. for all paths* $p$ *of* $r$ *we have that*

$$Inf(L|_p) \cap F \neq \emptyset.$$

$L|_p$ *denotes the set of states in* $L$ *which do also appear on* $p$*.*

**Definition 4.35 (Rabin tree automaton).** *A Rabin tree automaton is given by an* $\omega$*-tree automaton* $A = (Q, q_I, \Delta, \Omega)$ *where*

$$\Omega = \{(L_1, U_1), \ldots, (L_n, U_n)\}$$

*where each pair* $(L_i, U_i) \subseteq Q \times Q$ *is a set of "accepting" pairs (these pairs are called Rabin pairs). A run* $r = (dom_r, L)$ *is* successful *if, and only if, for each path* $p$ *on* $r$ *there is an index* $i \in \{1, \ldots, n\}$ *such that no state (resp. a state) from* $L_i$ *(resp. from* $U_i$*) occurs infinitely often on* $p$*; i.e.*

$$Inf(L|_p) \cap L_i = \emptyset \quad \text{and} \quad Inf(L|_p) \cap U_i \neq \emptyset$$

One can easily see that any set of trees acceptable by a Büchi tree automaton is also acceptable by a Rabin tree automaton (one takes as Rabin pairs the set $\{(\emptyset, F)\}$ where $F$ is the set of final states of the Büchi tree automaton). However, the converse is not true.

**Theorem 4.36 ([Rabin, 1970]).** *There is a set of trees that is acceptable by a Rabin tree automaton but not by any Büchi tree automaton.*

**Theorem 4.37 ([Rabin, 1970; Vardi and Wolper, 1984]).** *The emptiness problem for Büchi tree automata is decidable and* **P***-complete under logarithmic space reductions.*

**Theorem 4.38 ([Emerson and Jutla, 1988; Pnueli and Rosner, 1989]).** *The non-emptiness problem for Rabin tree automata is decidable and complete for* **NP**.

## 4.2 An Argumentative Approach to Coalition Formation

In this section we present an argument-based characterisation of coalition formation that will be used later to extend **ATL**. We follow the approach from [Amgoud, 2005a], where the argumentation framework for generating coalition structures is defined. The approach is a generalisation of the framework of Dung for argumentation [Dung, 1995], extended with a *preference relation*. The basic notion is that of a *coalitional framework*, which contains a set of elements $\mathfrak{C}$ (usually seen as agents or coalitions), an attack relation (for modelling conflicts among elements of $\mathfrak{C}$), and a preference relation between elements of $\mathfrak{C}$ (to describe favourite agents/coalitions).

**Definition 4.39 (Coalitional framework [Amgoud, 2005a]).** *A* coalitional framework *is a triple $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ where $\mathfrak{C}$ is a non-empty set of elements, $\mathcal{A} \subseteq \mathfrak{C} \times \mathfrak{C}$ is an attack relation, and $\prec$ is a preorder on $\mathfrak{C}$ representing preferences on elements in $\mathfrak{C}$.*

*Let $S$ be a non-empty set of elements. $\mathbb{CF}(S)$ denotes the set of all coalitional frameworks where elements are taken from the set $S$, i.e. for each $(\mathfrak{C}, \mathcal{A}, \prec) \in \mathbb{CF}(S)$ we have that $\mathfrak{C} \subseteq S$.*

The set $\mathfrak{C}$ in Definition 4.39 is intentionally generic, accounting for various possible alternatives. One alternative is to consider $\mathfrak{C}$ as a set of agents $\mathbb{A}\text{gt} = \{1, \ldots, k\}$: $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec) \in \mathbb{CF}(\mathbb{A}\text{gt})$. Then, a *coalition* is given by $C = \{i_1, \ldots, i_l\} \subseteq \mathfrak{C}$ and "agent" can be used as an intuitive reference to elements of $\mathfrak{C}$. Another alternative is to use a coalitional framework $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ based on $\mathbb{CF}(\mathcal{P}(\mathbb{A}\text{gt}))$. Now elements of $\mathfrak{C} \subseteq \mathcal{P}(\mathbb{A}\text{gt})$ are *groups* or *coalitions* (where we consider singletons as groups too) of agents. Under this interpretation a coalition $C \subseteq \mathfrak{C}$ is a *set of sets* of agents. Although "coalition" is already used for $C \subseteq \mathfrak{C}$, we also use the intuitive reading "coalition" or "group" to address elements in $\mathfrak{C}$.[1] Yet another way is not to use the specific structure for elements in $\mathfrak{C}$, assuming it just consists of abstract elements, e.g. $c_1$, $c_2$, etc. One may think of these elements as individual agents or coalitions. This approach is followed in [Amgoud, 2005a].

In the rest of this paper we mainly follow the first alternative when informally speaking about coalitional frameworks, i.e. we consider $\mathfrak{C}$ as a set of agents.

*Example 4.40.* Consider the following two coalitional frameworks: (i) $\mathcal{CF}_1 = (\mathfrak{C}, \mathcal{A}, \prec)$ where $\mathfrak{C} = \{a_1, a_2, a_3\}$, $\mathcal{A} = \{(a_3, a_2), (a_2, a_1), (a_1, a_3)\}$ and agent $a_3$ is preferred over $a_1$, i.e. $a_1 \prec a_3$; and (ii) $\mathcal{CF}_2 = (\mathfrak{C}', \mathcal{A}', \prec')$ where $\mathfrak{C}' = \{\{a_1\}, \{a_2\}, \{a_3\}\}$, $\mathcal{A}' = \{(\{a_3\}, \{a_2\}), (\{a_2\}, \{a_1\}), (\{a_1\}, \{a_3\})\}$ and

[1] The first interpretation is a special case of the second (coalitional frameworks are members $\mathbb{CF}(\mathcal{P}(\mathbb{A}\text{gt}))$).


**Fig. 4.1.** Figure (a) (resp. (b)) corresponds to the coalitional frameworks defined in Example 4.40 (resp. 4.49 (b)). Nodes represent agents and arrows between nodes stand for the attack relation.

group $\{a_3\}$ is preferred over $\{a_1\}$, i.e. $\{a_1\} \prec' \{a_3\}$. They capture the same scenario and are isomorphic but $\mathcal{CF}_1 \in \mathbb{CF}(\{a_1, a_2, a_3\})$ and $\mathcal{CF}_2 \in \mathbb{CF}(\mathcal{P}(\{a_1, a_2, a_3\}))$; that is, the first framework is defined regarding single agents and the latter over (trivial) coalitions. Figure 4.1 (a) shows a graphical representation of the first coalitional framework.

Let $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ be a coalitional framework. For $C, C' \in \mathfrak{C}$, we say that $C$ *attacks* $C'$ iff $C \mathcal{A} C'$. The attack relation represents conflicts between elements of $\mathfrak{C}$; for instance, two agents may rely on the same (unique) resource or they may have disagreeing goals, which prevent them from cooperation. However, the notion of attack may not be sufficient for modelling conflicts, as some elements (resp. coalitions) in $\mathfrak{C}$ may be preferred over others. This leads to the notion of *defeater* which combines the notions of attack and preference.

**Definition 4.41 (Defeater).** *Let $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ be a coalitional framework and let $C, C' \in \mathfrak{C}$. We say that $C$* defeats *$C'$ if, and only if, $C$ attacks $C'$ and $C'$ is not preferred over $C$ (i.e., not $C \prec C'$). We also say that $C$ is a* defeater *for $C'$.*

Attacks and defeats are defined between *single* elements of $\mathfrak{C}$. As we are interested in the formation of coalitions it is reasonable to consider conflicts between coalitions. Members in a coalition may prevent attacks to members in the same coalition; they protect each other. The concept of defence, introduced next, captures this idea of mutual protection.

**Definition 4.42 (Defence).** *Let $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ be a coalitional framework and $C, C' \in \mathfrak{C}$. We say that $C'$* defends itself against *$C$ if, and only if, $C'$ is preferred over $C$, i.e., $C \prec C'$, and $C'$* defends itself *if it defends itself against any of its attackers. Furthermore, $C$ is* defended by *a set $\mathfrak{S} \subseteq \mathfrak{C}$ of elements of $\mathfrak{C}$ if, and only if, for all $C'$ defeating $C$ there is a coalition $C'' \in \mathfrak{S}$ defeating $C'$.*

In other words, if an element $C'$ defends itself against $C$ then $C$ may attack $C'$ but $C$ is not allowed to defeat $C'$.

A minimal requirement one should impose on a coalition is that its members do not defeat each other; otherwise, the coalition may be unstable and break up sooner or later because of conflicts among its members. This is formalised in the next definition.

**Definition 4.43 (Conflict-free).** *Let $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ be a coalitional framework and $\mathfrak{S} \subseteq \mathfrak{C}$ a set of elements in $\mathfrak{C}$. Then, $\mathfrak{S}$ is called* conflict-free *if, and only if, there is* no *$C \in \mathfrak{S}$ defeating some member of $\mathfrak{S}$.*

It must be remarked that our notions of "defence" and "conflict-free" are defined in terms of "defeat" rather than "attack".[2] Given a coalitional framework $\mathcal{CF}$ we will use argumentation to compute coalitions with desirable properties. In argumentation theory many different semantics have been proposed to define ultimately accepted arguments [Dung, 1995; Caminada, 2006]. We apply this rich framework to provide different ways to coalition formation. A semantics can be defined as follows.

**Definition 4.44 (Coalitional framework semantics).** *A* semantics *for a coalitional framework $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ is a (isomorphism invariant) mapping $\mathfrak{sem}$ which assigns to a given coalitional framework $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ a set of subsets of $\mathfrak{C}$, i.e., $\mathfrak{sem}(\mathcal{CF}) \subseteq \mathcal{P}(\mathfrak{C})$.*

Let $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ be a coalitional framework. To formally characterise different semantics we will define a function $\mathcal{F}_{\mathcal{CF}} : \mathcal{P}(\mathfrak{C}) \to \mathcal{P}(\mathfrak{C})$ which assigns to a set of coalitions $\mathfrak{S} \in \mathcal{P}(\mathfrak{C})$ the coalitions defended by $\mathfrak{S}$.

**Definition 4.45 (Characteristic function $\mathcal{F}$).** *Let $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ be a coalitional framework and $\mathfrak{S} \subseteq \mathfrak{C}$. The function $\mathcal{F}$ defined by*

$$\mathcal{F}_{\mathcal{CF}} : \mathcal{P}(\mathfrak{C}) \to \mathcal{P}(\mathfrak{C})$$
$$\mathcal{F}_{\mathcal{CF}}(\mathfrak{S}) = \{C \in \mathfrak{C} \mid C \textit{ is defended by } \mathfrak{S}\}$$

*is called* characteristic function.[3]

$\mathcal{F}$ can be applied recursively to coalitions resulting in new coalitions. For example, $\mathcal{F}(\emptyset)$ provides all undefeated coalitions and $\mathcal{F}^2(\emptyset)$ constitutes the set of all elements of $\mathfrak{C}$ which members are undefeated *or* are defended by undefeated coalitions.

*Example 4.46.* Consider again the coalitional framework $\mathcal{CF}_1$ given in Example 4.40. The characteristic function applied on the empty set results in $\{a_3\}$ since the agent is undefeated, $\mathcal{F}(\emptyset) = \{a_3\}$. Applying $\mathcal{F}$ on $\mathcal{F}(\emptyset)$ determines the set $\{a_1, a_3\}$ because $a_1$ is defended by $a_3$. It is easy to see that $\{a_1, a_3\}$ is a fixed-point of $\mathcal{F}$.

---

[2] In [Amgoud, 2005a,b] these notions are defined the other way around, resulting in a different characterisation of stable semantics.

[3] We omit the subscript $\mathcal{CF}$ if it is clear from context.

We now introduce the first concrete semantics called coalition structure semantics, which was originally defined in [Amgoud, 2005a].

**Definition 4.47 (Coalition structure $\mathfrak{sem}_{\mathbf{cs}}$ [Amgoud, 2005a]).** *Let* $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ *be a coalitional framework. Then*

$$\mathfrak{sem}_{cs}(\mathcal{CF}) := \left\{ \bigcup_{i=1}^{\infty} \mathcal{F}^{i}_{\mathcal{CF}}(\emptyset) \right\}$$

*is called* coalition structure semantics *or just* coalition structure *for* $\mathcal{CF}$*.*

For a coalitional framework $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ with a finite set $\mathfrak{C}$[4] the characteristic function $\mathcal{F}$ is continuous [Dung, 1995, Lemma 28]. Since $\mathcal{F}$ is also monotonic it has a least fixed-point given by $\mathcal{F}(\emptyset)\uparrow^{\omega}$ (according to Knaster-Tarski). We have the following straightforward properties of coalition structure semantics.

**Proposition 4.48 (Coalition structure).** *Let* $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ *be a coalitional framework with a finite set* $\mathfrak{C}$*. There is always a unique coalition structure for* $\mathcal{CF}$*. Furthermore, if no element of* $C \in \mathfrak{C}$ *defends itself then the coalitional structure is empty, i.e.* $\mathfrak{sem}_{cs}(\mathcal{CF}) = \{\emptyset\}$*.*

*Example 4.49.* The following situations illustrate the notion of coalitional structure:

(a) Consider Example 4.46. Since $\{a_1, a_3\}$ is a fixed-point of $\mathcal{F}_{\mathcal{CF}_1}$ the coalitional framework $\mathcal{CF}_1$ has $\{a_1, a_3\}$ as coalitional structure.
(b) $\mathcal{CF}_3 := (\mathfrak{C}, \mathcal{A}, \prec) \in \mathbb{CF}(\{a_1, a_2, a_3\})$ (shown in Figure 4.1(b)), is a coalitional framework with $\mathfrak{C} = \{a_1, a_2, a_3\}$, $\mathcal{A} = \{\ (a_1, a_2),\ (a_1, a_3)\ ,\ (a_2, a_1),\ (a_2, a_3),\ (a_3, a_1)\}$ and $a_3$ is preferred over $a_2$, $a_2 \prec a_3$, has the empty coalition as associated coalition structure, i.e. $\mathfrak{sem}_{\mathrm{cs}}(\mathcal{CF}) = \{\emptyset\}$.

Since the coalition structure is often very restrictive, it seems reasonable to introduce other less restrictive semantics. Each of the following semantics are well-known in argumentation theory [Dung, 1995] and can be used as a criterion for coalition formation (cf. [Amgoud, 2005a]).

**Definition 4.50 (Argumentation semantics).** *Let* $(\mathfrak{C}, \mathcal{A}, \prec)$ *be a coalitional framework,* $\mathfrak{S} \subseteq \mathfrak{C}$ *a set of elements of* $\mathfrak{C}$*.* $\mathfrak{S}$ *is called*

*(a)* admissible extension *iff* $\mathfrak{S}$ *is conflict-free and* $\mathfrak{S}$ *defends all its elements, i.e.* $\mathfrak{S} \subseteq \mathcal{F}(\mathfrak{S})$*.*
*(b)* complete extension *iff* $\mathfrak{S}$ *is conflict-free and* $\mathfrak{S} = \mathcal{F}(\mathfrak{S})$*.*

[4] Actually, it is enough to assume that $\mathcal{CF}$ is finitary (cf. [Dung, 1995, Def. 27]).

*(c)* grounded extension *iff $\mathfrak{S}$ is the smallest (wrt. to set inclusion) complete extension.*

*(d)* preferred extension *iff $\mathfrak{S}$ is a maximal (wrt. to set inclusion) admissible extension.*

*(e)* stable extension *iff $\mathfrak{S}$ is conflict-free and it defeats all arguments not in $\mathfrak{S}$.*

*Let $\mathfrak{sem}_{cs}$ (resp. $\mathfrak{sem}_{complete}$, $\mathfrak{sem}_{grounded}$, $\mathfrak{sem}_{preferred}$ and $\mathfrak{sem}_{stable}$) denote the semantics which assigns to a coalitional structure $\mathcal{CF}$ all its admissible (resp. complete, grounded, preferred, and stable) extensions.*

*Remark 4.51.* We note that the grounded extension is equivalent to the coalition structure semantics.

There is only one unique coalition structure (possibly the empty one) for a given coalitional framework, but there can be several stable and preferred extensions. The existence of at least one preferred extension is guaranteed which is not the case for the stable semantics. Thus, the possible coalitions very much depend on the used semantics.

*Example 4.52.* For $\mathcal{CF}_3$ from Example 4.49 the following holds:

$$\begin{aligned}
&\mathfrak{sem}_{\text{cs}}(\mathcal{CF}) = \mathfrak{sem}_{\text{grounded}}(\mathcal{CF}) = \{\emptyset\} \\
&\mathfrak{sem}_{\text{admissible}}(\mathcal{CF}) = \{\emptyset, \{a_1\}, \{a_2\}, \{a_3\}, \{a_2, a_3\}\} \\
&\mathfrak{sem}_{\text{complete}}(\mathcal{CF}) = \{\emptyset, \{a_1\}, \{a_2, a_3\}\} \\
&\mathfrak{sem}_{\text{preferred}}(\mathcal{CF}) = \mathfrak{sem}_{\text{stable}}(\mathcal{CF}) = \{\{a_1\}, \{a_2, a_3\}\}
\end{aligned}$$

Analogously, for the coalitional framework $\mathcal{CF}_1$ from Example 4.40 there exists one complete extension $\{a_1, a_3\}$ which is also a grounded, preferred, and stable extension.

## 4.3 Probability Theory

In this section we recall some basic notions from probability theory. Let $X$ be a non-empty set and let $\mathcal{F} \subseteq \mathcal{P}(X)$ be a set of subsets. $\mathcal{F}$ is called a *(set) algebra over $X$* iff: (i) $\emptyset \in \mathcal{F}$; (ii) if $A \in \mathcal{F}$ then also $\bar{A} := X \setminus A \in \mathcal{F}$; (iii) if $A, B \in \mathcal{F}$ then also $A \cup B \in \mathcal{F}$. $\mathcal{F}$ is called *$\sigma$-algebra* iff also (iv) $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$ for all $A_1, A_2, \dots \in \mathcal{F}$.

Let $\mathcal{S}$ be a $\sigma$-algebra over $X$. We say that a function $\mu : \mathcal{S} \to \mathbb{R}$ is a *measure (on $\mathcal{S}$)* iff it is non-negative, i.e. $\mu(A) \geq 0$ for all $A \in \mathcal{S}$, and $\sigma$-additive, i.e. $\mu(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mu(A_i)$ whenever $\bigcup_{i=1}^{\infty} A_i \in \mathcal{S}$ and all $A_i$ pairwise disjoint. We note that these properties imply that $\mu(\emptyset) = 0$.

Finally, we say that the measure $\mu$ is a *probability measure* if $\mu(X) = 1$ and call the triple $(X, \mathcal{S}, \mu)$ a *probability space*. By $\Xi(\mathcal{S})$ we denote the set of all probability measures over $\mathcal{S}$.

Note that whenever $X$ is finite it is sufficient to define the probabilities of the basic elements $x \in X$. Then, the probability of an event $E \subseteq X$ is given by the sum of the basic probabilities: $\mu(E) = \sum_{x \in E} \mu(\{x\})$, and the corresponding probability measure is uniquely determined over the $\sigma$-algebra $\mathcal{P}(X)$. In such cases, we can also write $\mu(x)$ instead of $\mu(\{x\})$ and $\Xi(X)$ instead of $\Xi(\mathcal{P}(X))$, and do also refer to a probability measure over $\mathcal{P}(X)$ as probability measure over $X$. The *support* of a probability measure $\mu \in \Xi(X)$ is the set of elements $x \in X$ such that $\mu(x) > 0$, denoted by

$$\mathsf{Supp}(\mu) = \{x \in X \mid \mu(x) > 0\}.$$

Next, we introduce a Markov decision process (MDP). The definition is based on [de Alfaro et al., 2004]. We note that in the literature there are subtle differences how rewards are included in MDPs.

**Definition 4.53 (Markov decision process).** *A (finite)* Markov decision process *(*MDP*) is defined as a tuple*

$$\mathcal{D} = (Q, \tau, \Pi, [\cdot])$$

*where*

- *$Q$ is a finite set of states.*
- *$\tau : Q \to \mathcal{P}(\Xi(Q))\backslash\emptyset$ a probabilistic transition relation. It assigns to each state a non-empty* set *of probability distributions over the set of states. Each element $a \in \tau(q)$ is an action that determines the transition probabilities to the next states.*
- *$\Pi$ is a set of propositions.*
- *$[\cdot] : \Pi \to (Q \to [0, 1])$ is a valuation function assigning to each proposition a state-depended reward.*

Similarly to CGSs the system may evolve in different ways resulting in an execution tree of infinite paths. We define a finite (resp. infinite) *trajectory* (or *path*) of $\mathcal{D}$ as a finite (resp. infinite) sequence $q_0 q_1 \ldots q_i \in Q^+$ (resp. $q_0 q_1 \ldots \in Q^\omega$) of states such that for all $j < i$ (resp. $j \geq 0$) there is an action $a_j \in \tau(q_j)$ with $q_j \in \mathsf{Supp}(a_j)$. We define *Traj* and *FTraj* as the infinite and finite trajectories and *Traj*$(q)$ as the subset of *Traj* that contains all trajectories starting in $q$.

**Definition 4.54 (Policy).** *A* policy *of an* MDP $\mathcal{D}$ *is a function*

$$\mathsf{pol} : FTraj \to \Xi(\bigcup_{q \in Q} \tau(q))$$

*such that* $\mathsf{Supp}(\mathsf{pol}(q_0 \dots q_i)) \subseteq \tau(q_i)$. *We denote by* $\mathsf{Pols}_\mathcal{D}$ *the set of all such policies over* $\mathcal{D}$.

We note that a policy is history dependent similarly to perfect recall strategies (*IR*-strategies) from Definition 2.17. For each finite history, a policy choses a probability over the available actions (which are themselves probability distributions over transitions).

Now, given $q_0 \dots q_i$ and a policy pol the probability that the next state will be $q_{i+1}$ is defined as follows:

$$\mathsf{next}_{\mathsf{pol}}(q_{i+1} \mid q_0 \dots q_i) := \sum_{a \in \tau(q_i)} \mathsf{pol}(q_0 \dots q_i)(a) \cdot a(q_{i+1}).$$

For an initial state $q \in Q$ we can define the trajectory probability space of a MDP wrt. a policy at hand.

**Definition 4.55 ($(\mathcal{D}, \mathsf{pol}, q)$-trajectory probability space).** *Let* $\mathcal{D}$ *be a* MDP, $\mathsf{pol} \in \mathsf{Pols}_\mathcal{D}$, *and* $s \in Q_\mathcal{D}$. *The* $(\mathcal{D}, \mathsf{pol}, q)$-trajectory probability space *is defined as*

$$\mathcal{T}_{\mathcal{D},\mathsf{pol},q} = (\mathit{Traj}(q), \mathcal{B}_q, P_q^{\mathsf{pol}})$$

*where*

- $\mathcal{B}_q$ *is the set of measurable subsets of* $\mathit{Traj}(q)$ *and*
- $P_q^{\mathsf{pol}}$ *is the probability measure over* $\mathcal{B}_q$ *induced by* $\mathsf{next}_{\mathsf{pol}}(\cdot)$.

*We use* $\mathbf{E}_q^{\mathsf{pol}}[X]$ *to denote the expected value of the random variable* $X$ *over* $\mathcal{T}_{\mathcal{D},\mathsf{pol},q}$.

## 4.4 Petri Nets

In the following we introduce Petri nets and some basic problems about them. The latter are required for the proofs in Section 11.2.

### 4.4.1 Basic Definitions

**Definition 4.56 (Petri net).** *A* Petri net *is a tuple*

$$N = (S, T, W, m^I)$$

*where*

- $S$ *and* $T$ *are non-empty and disjoint sets of places and transitions, respectively;*

- $W : (S \times T) \cup (T \times S) \to \mathbb{N}_0$ *represents arc weights that determine how many tokens are needed by and produced by each transition; and*
- $m^I : S \to \mathbb{N}_0$ *is the initial marking, i.e., a distribution of tokens on the places in the net.*

The *state* (also called *marking*) of a Petri net is defined as the distribution of tokens. Given a state some transitions are activated namely those transitions $t$ for which sufficient tokens are available in the respective places.

**Definition 4.57 (State, enabled, subsequent marking, firing).** *Let* $N = (S, T, W, m^I)$ *be a Petri net.*

*(a) A* state *or* marking *of* $N$ *is a function* $m : S \to \mathbb{N}_0$.
*(b) A transition* $t \in T$ *is* enabled by a marking $m$, *denoted by* $m[t\rangle$, *iff* $m(s) \geq W(s, t)$ *for all* $s \in S$.
*(c) If* $t$ *is enabled by* $m$ *we say that* $m'$ *is a* subsequent state, *denoted* $m[t\rangle m'$, *if* $m'(s) = m(s) - W(s, t) + W(t, s)$ *for all* $s \in S$. *We also say that* $t$ fires *in* $m$ *and yields* $m'$.

For a sequence $\sigma = t_1 \dots t_n \in T^*$ of subsequently enabled transitions we can lift the notion of firing, we write $m[\sigma\rangle$ and $m[\sigma\rangle m'$, respectively. Given the initial state $m$ each subsequence $t_1 \dots t_i$ of $\sigma$ with $i < n$ determines a unique state referred to as $m_i$. Then, we do also write

$$\sigma_m := m t_1 m_1 \dots t_i m_i.$$

Later, we shall be interested whether a Petri net can fire infinitely often. This corresponds to an *infinite* sequence of subsequently firing transitions.

**Definition 4.58 ($m$-run).** *Let* $N$ *be a Petri net and* $m$ *a state in it. An* $m$-run *in a Petri net* $N$ *is a infinite sequence* $\sigma = t_1 t_2 \dots \in T^\omega$ *of subsequently enabled transitions; that is, for each* $i \geq 1$ *we have that* $m[t_1 \dots t_i\rangle$.

### 4.4.2 Reachability Problems

A well-known and computationally complex problem for Petri nets is the *reachability problem.* The problem is characterised by the question whether some state $m_2$ is reachable from a state $m_1$. That is, we are after the existence of a (finite) firing sequence $\sigma$ such that $m_1[\sigma\rangle m_2$. Formally, the reachability problem *Reach* is given by the following set

$$\{(N, m_1, m_2) \mid N = (S, T, W, m_1),\ m_1, m_2 : S \to \mathbb{N}_0, \exists \sigma \in T^* \ (m_1[\sigma\rangle m_2)\}.$$

The problem *ExtReach* (*extended reachability problem*) is an extension of the reachability problem and asks for a run along which specific states occur infinitely often. In Section 11.2 we will use this problem to show that the resource-bounded logic **RTL** is decidable.

**Definition 4.59 (Extended Reachability [Jančar, 1990, Def. 2.9]).** *Let a Petri net* $N = (S, T, W, m^I)$ *and a pair* $(A, f)$ *such that* $A \subseteq S$ *and* $f : A \to \mathbb{N}_0$ *be given. The* extended reachability problem *ExtReach is given as follows:*

> *Is there an* $m^I$*-run* $\sigma = t_1 t_2 \ldots$ *such that there are infinitely many indices* $i$ *such that the marking* $m_i$ *that occurs after* $t_i$ *restricted to the states in* $A$ *equals* $f$ *(i.e.,* $m_i \restriction_A \equiv f$ *for infinitely many* $i$*)?*

**Theorem 4.60 ([Jančar, 1990]).** *The problem ExtReach is decidable.*

# 5

# Model Checking Temporal and Strategic Logics

## Contents



Model checking is a powerful method which can for instance be used for the verification of computer systems. Given a model and a formula in a certain logic, model checking determines whether the formula is true in the model. Often, it is used to check specifications of desirable properties a system should fulfil. If the formula is true, we know that the property expressed by the formula is satisfied in the model. If not, it might lead us to change the system or at least gives hints how to debug it. The advantage of model checking over other methods like *simulation and testing* (see for instance [Myers and Sandler, 2004]) and deductive reasoning (Hoare calculus) is that it is usually decidable; albeit, there are exceptions to this.

Model checking was invented and pioneered by the work of Edward Melson Clarke, Ernest Allen Emerson, and by Joseph Sifakis and Jean Pierre Queille in the 80ies as a means for formal verification of finite-state concurrent systems. Specifications about the system were expressed as temporal logic formulae. It was especially suited for checking hardware designs, but also applied to checking software specifications. While it started as a new approach replacing the then common Floyd-Hoare style logic, it could only handle relatively small (though non-trivial) examples. Scalability was an important motivation right from the beginning. The last years have seen many industrial applications, and a number of powerful model checkers are available today. As founders of a new and flourishing area in computer science, Clarke, Emerson and Sifakis have been honoured with the Turing award in 2007.

Logic-based verification of multi-agent systems has become an important subfield on its own. Some important model checkers are:

- Mocha [Alur et al., 1998a], available for download at http://www.cis.upenn.edu/ mocha/,
- VeriCS [Dembiński et al., 2003], available at http://pegaz.ipipan.waw.pl/verics/,
- MCMAS [Raimondi and Lomuscio, 2004; Raimondi, 2006], available at http://www-lai.doc.ic.ac.uk/mcmas/.

In this chapter, we do not deal with practical aspects of MASs verification. Instead, we offer a comprehensive survey of theoretical results concerning the computational complexity of model checking for relevant properties of agents and their teams. To this end, we focus on the class of properties that can be specified in the temporal and strategic logics presented in Chapter 2.

Naturally, there is more than model checking to be studied. In [Goranko and van Drimmelen, 2003] a complete *axiomatisation* for $\mathsf{ATL_{IR}}$ is presented. Also the *satisfiability problem* of $\mathsf{ATL_{IR}}$ and $\mathsf{ATL^*_{IR}}$ has been considered by researchers: The problem was proven **EXPTIME**-complete for $\mathsf{ATL_{IR}}$ [van Drimmelen, 2003; Walther et al., 2006] and **2EXPTIME**-complete for $\mathsf{ATL^*_{IR}}$ [Schewe, 2008]. Axiomatisation and satisfiability of other variants of alternating-time temporal logic still remain open.

## 5.1 The Model Checking Problem

Model checking is a technique for verifying *finite state* systems. Naturally this is achieved in a fully automated way which is an advantage over other methods like simulation and testing and deductive reasoning. The latter has the advantage that it can also cope with infinite state systems but usually this cannot be achieved fully automatically.

The process of model checking seeks to answer the question whether a given formula $\varphi$ is satisfied in a state $q$ of model $\mathfrak{M}$. Formally, *local model checking* is the decision problem that determines membership in the set

$$\mathsf{MC}(\mathcal{L}, \mathsf{Struc}, \models) := \{(\mathfrak{M}, q, \varphi) \in \mathsf{Struc} \times \mathcal{L} \mid \mathfrak{M}, q \models \varphi\},$$

where $\mathcal{L}$ is a logical language, Struc is a class of (pointed) models for $\mathcal{L}$ (i.e. a tuple consisting of a model and a state), and $\models$ is a semantic satisfaction relation compatible with $\mathcal{L}$ and Struc. We omit parameters if they are clear from context, e.g., we use MC(**CTL**) to refer to model checking of $\mathcal{L}_{CTL}$ over the class of (pointed) Kripke models and the introduced semantics.

It is often useful to compute the set of states in $\mathfrak{M}$ that satisfy formula $\varphi$ instead of checking if $\varphi$ holds in a particular state. This variant of the problem is known as *global model checking*. It is easy to see that, for the settings we consider here, the complexities of local and global model checking coincide, and the algorithms for one variant of model checking can be adapted to the other variant in a simple way. As a consequence, we will use both notions of model checking interchangeably.

In the following, we are interested in the decidability and the computational complexity of determining whether an input instance $(\mathfrak{M}, q, \varphi)$ belongs to MC(...). The complexity is always relative to the *size* of the instance; in the case of model checking, it is the size of the representation of the model and the representation of the formula that we use. Thus, in order to establish the complexity, it is necessary to fix how we *represent* the input and how we *measure* its size. In the following sections, we firstly consider *explicit representation* of models and formulae, together with the "standard" input measure, where the size of the model ($|\mathfrak{M}|$) is given by the *number of transitions* in $\mathfrak{M}$, and the size of the formula ($|\varphi|$) is given by its *length* (i.e., the number of elements it is composed of, apart from parentheses). For example, the model in Figure 2.2 includes 12 (labeled) transitions, and the formula $\langle\!\langle 1 \rangle\!\rangle \bigcirc (\mathsf{pos}_0 \vee \mathsf{pos}_1)$ has length 5.

## 5.2 Linear- and Branching Time Logics: LTL, CTL$^\star$, and CTL

An excellent survey on the model checking complexity of temporal logics has been presented in [Schnoebelen, 2003]. Here, we only recall the results relevant for the subsequent analysis of strategic logics.

Let $\mathfrak{M}$ be a Kripke model and $q$ be a state in the model. Model checking an $\mathcal{L}_{CTL}/\mathcal{L}_{CTL^*}$-formula $\varphi$ in $\mathfrak{M}, q$ means to determine whether $\mathfrak{M}, q \models \varphi$, i.e., whether $\varphi$ holds in $\mathfrak{M}, q$. For **LTL**, checking $\mathfrak{M}, q \models \varphi$ means that we check the *validity* of $\varphi$ in the pointed model $\mathfrak{M}, q$, i.e., whether $\varphi$ holds *on*

| **function** $mcheck(\mathfrak{M}, \varphi)$. |
|---|
| Model checking formulae of CTL. Returns the exact subset of $Q$ for which formula $\varphi$ holds. |
| **case** $\varphi \equiv p$ **:** Â return $\{q \in Q \mid p \in \pi(q)\}$<br>**case** $\varphi \equiv \neg\psi$ **:** return $Q \setminus mcheck(\mathfrak{M}, \psi)$<br>**case** $\varphi \equiv \psi_1 \wedge \psi_2$ **:** return $mcheck(\mathfrak{M}, \psi_1) \cap mcheck(\mathfrak{M}, \psi_2)$<br>**case** $\varphi \equiv \mathsf{E} \bigcirc \psi$ **:** return $pre(mcheck(\mathfrak{M}, \psi))$<br>**case** $\varphi \equiv \mathsf{E}\Box\psi$ **:**<br>  $Q_1 := Q$;  $Q_2 := Q_3 := mcheck(\mathfrak{M}, \psi)$;<br>  **while** $Q_1 \not\subseteq Q_2$ **do** $Q_1 := Q_1 \cap Q_2$; $Q_2 := pre(Q_1) \cap Q_3$ **od**;<br>  return $Q_1$<br>**case** $\varphi \equiv \mathsf{E}\psi_1\mathcal{U}\psi_2$ **:**<br>  $Q_1 := \emptyset$;  $Q_2 := mcheck(\mathfrak{M}, \psi_2)$;  $Q_3 := mcheck(\mathfrak{M}, \psi_1)$;<br>  **while** $Q_2 \not\subseteq Q_1$ **do** $Q_1 := Q_1 \cup Q_2$; $Q_2 := pre(Q_1) \cap Q_3$ **od**;<br>  return $Q_1$<br>**end case** |

**Fig. 5.1.** The **CTL** model checking algorithm from [Clarke and Emerson, 1981].

*all the paths* in $\mathfrak{M}$ that start from $q$ (equivalent to **CTL**$^\star$ model checking of formula $\mathsf{A}\varphi$ in $\mathfrak{M}, q$, cf. Remark 2.11).

It has been known for a long time that formulae of **CTL** can be model checked in time linear with respect to the size of the model and the length of the formula [Clarke et al., 1986], whereas formulae of **LTL** and **CTL**$^\star$ are significantly harder to verify.

**Theorem 5.1 (CTL [Clarke et al., 1986; Schnoebelen, 2003]).** *Model checking* **CTL** *is* **P***-complete, and can be done in time* $\mathbf{O}(|\mathfrak{M}| \cdot |\varphi|)$*, where* $|\mathfrak{M}|$ *is given by the number of transitions.*

*Proof.* [Sketch] The algorithm determining the states in a model at which a given formula holds is presented in Figure 5.1. The lower bound (**P**-hardness) can for instance be proven by a reduction of the Circuit-value problem [Schnoebelen, 2003]. ■

**Theorem 5.2 (LTL [Sistla and Clarke, 1985; Lichtenstein and Pnueli, 1985; Vardi and Wolper, 1986]).** *Model checking* **LTL** *is* **PSPACE***-complete, and can be done in time* $2^{\mathbf{O}(|\varphi|)}\mathbf{O}(|\mathfrak{M}|)$*, where* $|\mathfrak{M}|$ *is given by the number of transitions.*

*Proof.* [Sketch] We sketch the approach given in [Vardi and Wolper, 1986]. Firstly, given an $\mathcal{L}_{LTL}$-formula $\varphi$, a Büchi automaton $\mathcal{A}_{\neg\varphi}$ of size $2^{\mathbf{O}(|\varphi|)}$ accepting exactly the paths satisfying $\neg\varphi$ is constructed. The pointed Kripke

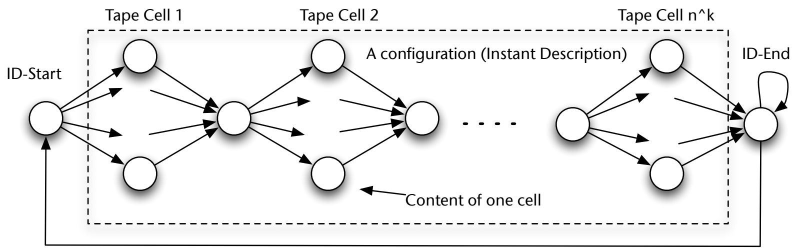



**Fig. 5.2.** Encoding of a $S(n)$-space bounded DTM as a Kripke model.

model $\mathfrak{M}, q$ can directly be interpreted as a Büchi automaton $\mathcal{A}_{\mathfrak{M},q}$ of size $\mathbf{O}(|\mathfrak{M}|)$ accepting all possible paths in the Kripke model starting in $q$. Then, the model checking problem reduces to the non-emptiness check of $L(\mathcal{A}_{\mathfrak{M},q}) \cap L(\mathcal{A}_{\neg\varphi})$ which can be done in time $\mathbf{O}(|\mathfrak{M}|) \cdot 2^{\mathbf{O}(|\varphi|)}$ by constructing the product automaton. (Emptiness can be checked in linear time wrt to the size of the automaton.) A **PSPACE**-algorithm is obtained by "guessing" a periodic paths that falsifies $\varphi$ (cf. Corollary 4.30). For details we refer e.g. to [Baier and Katoen, 2008].

A **PSPACE**-hardness proof can for instance be found in [Sistla and Clarke, 1985]. The idea is to simulate the computation of a polynomially space-bounded deterministic TM as a Kripke model and to use **LTL**-formulae to ensure that the machine is simulated correctly (cf. Figure 5.2). States of the model are taken as the symbols of the input alphabet of the TM, tuples $(s, a)$ where $s$ (resp. $a$) is a state of the machine (resp. input symbol), and two fresh states indicating the start and end of a configuration, respectively. A formula $\varphi$ is used to describe the initial and final configuration and to ensures that subsequent configuration are valid. Then, we have that the TM halts on input $w$ iff $\mathfrak{M}, q_0 \models \varphi$. ∎

The hardness of **CTL**$^\star$ model checking is immediate from Theorem 5.2 as $\mathcal{L}_{LTL}$ can be seen as a fragment of $\mathcal{L}_{CTL^*}$. For the proof of the upper bound one combines the **CTL** and **LTL** model checking techniques. We consider an $\mathcal{L}_{CTL^*}$-formula $\varphi$ which contains a state subformula $\mathsf{E}\psi$ where $\psi$ is a pure $\mathcal{L}_{LTL}$-formula. Firstly, we can use **LTL** model checking to determine all states which satisfy $\mathsf{E}\psi$ (these are all states $q$ in which the $\mathcal{L}_{LTL}$-formula $\neg\psi$ is *not* true) and label them by a fresh propositional symbol , say $\mathsf{p}$, and replace $\mathsf{E}\psi$ in $\varphi$ by $\mathsf{p}$ as well. Applying this procedure recursively yields a pure $\mathcal{L}_{CTL}$-formula which can be verified in polynomial time. Hence, the procedure can be implemented by an oracle machine of type $\mathbf{P}^{\mathbf{PSPACE}} = \mathbf{PSPACE}$ (the **LTL** model checking algorithm might be employed polynomially many times). Thus, the complexity for **CTL**$^\star$ is the same as for **LTL**.

**Theorem 5.3 ($\mathbf{CTL}^\star$ [Clarke et al., 1986; Emerson and Lei, 1987]).** *Model checking $\mathbf{CTL}^\star$ is* **PSPACE**-*complete, and can be done in time $2^{\mathbf{O}(|\varphi|)}\mathbf{O}(|\mathfrak{M}|)$, where $|\mathfrak{M}|$ is given by the number of transitions.*

In Section 2.2.5 we introduced $\mathbf{ATL}^+$, a variant of $\mathbf{ATL}$. As the model checking algorithm for $\mathbf{ATL}^+$ will rely on the complexity of $\mathbf{CTL}^+$ model checking, we mention the latter result here.

**Theorem 5.4 ($\mathbf{CTL}^+$ [Laroussinie et al., 2001]).** *Model checking $\mathbf{CTL}^+$ is $\mathbf{\Delta_2^P}$-complete in the number of transitions in the model and the length of the formula.*

## 5.3 Alternating Time Temporal Logics

In the following we consider the model checking problems for the strategic logics based on ATLs.

### 5.3.1 ATL and CL: Perfect Information

One of the main results concerning **ATL** states that its formulae can also be model checked in deterministic linear time, analogously to **CTL**. It is important to emphasise, however, that the result is relative to the number of transitions in the model and the length of the formula. In Section 5.4.1 we discuss an alternative input measure in terms of agents, states, and the length of the formula, and show that this causes a substantial increase in complexity.

The **ATL** model checking algorithm from [Alur et al., 2002] is presented in Figure 5.3. The algorithm employs the well-known fixedpoint characterisations of strategic-temporal modalities:

$$\langle\!\langle A\rangle\!\rangle\Box\varphi \leftrightarrow \varphi \wedge \langle\!\langle A\rangle\!\rangle \bigcirc \langle\!\langle A\rangle\!\rangle\Box\varphi,$$
$$\langle\!\langle A\rangle\!\rangle \varphi_1\mathcal{U}\varphi_2 \leftrightarrow \varphi_2 \vee \varphi_1 \wedge \langle\!\langle A\rangle\!\rangle \bigcirc \langle\!\langle A\rangle\!\rangle \varphi_1\mathcal{U}\varphi_2,$$

and computes a winning strategy step by step (if it exists). That is, it starts with the appropriate candidate set of states ($\emptyset$ for $\mathcal{U}$ and the whole set $Q$ for $\Box$), and iterates backwards over $A$'s one-step abilities until the set gets stable. It is easily seen that the algorithm needs to traverse each transition at most once per subformula of $\varphi$. It does not matter whether perfect recall or memoryless strategies are used: The algorithm is correct for the *IR*-semantics, but it always finds an *Ir*-strategy. Thus, for the $\mathcal{L}_{ATL}$-formula $\langle\!\langle A\rangle\!\rangle\gamma$, if $A$ have an *IR*-strategy to enforce $\gamma$, they also have an *Ir*-strategy to obtain it.

**Theorem 5.5 ($\mathbf{ATL_{Ir}}$ and $\mathbf{ATL_{IR}}$ [Alur et al., 2002]).** *Model checking $\mathbf{ATL_{Ir}}$ and $\mathbf{ATL_{IR}}$ is* **P**-*complete, and can be done in time $\mathbf{O}(|\mathfrak{M}| \cdot |\varphi|)$, where $|\mathfrak{M}|$ is given by the number of transitions in $\mathfrak{M}$.*

**function** $mcheck(M, \varphi)$.

**ATL** model checking. Returns the set of states in model $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, o\rangle$ for which formula $\varphi$ holds.

**case** $\varphi \in \Pi$ **:** return $\pi(p)$
**case** $\varphi = \neg\psi$ **:** return $Q \setminus mcheck(M, \psi)$
**case** $\varphi = \psi_1 \vee \psi_2$ **:** return $mcheck(M, \psi_1) \cup mcheck(M, \psi_2)$
**case** $\varphi = \langle\!\langle A \rangle\!\rangle \bigcirc \psi$ **:** return $pre(M, A, mcheck(M, \psi))$
**case** $\varphi = \langle\!\langle A \rangle\!\rangle \Box \psi$ **:**
  $Q_1 := Q$;  $Q_2 := mcheck(M, \psi)$;  $Q_3 := Q_2$;
  **while** $Q_1 \not\subseteq Q_2$
  **do** $Q_1 := Q_2$;  $Q_2 := pre(M, A, Q_1) \cap Q_3$ **od**;
  return $Q_1$
**case** $\varphi = \langle\!\langle A \rangle\!\rangle \psi_1 \mathcal{U} \psi_2$ **:**
  $Q_1 := \varnothing$;  $Q_2 := mcheck(M, \psi_1)$;
  $Q_3 := mcheck(M, \psi_2)$;
  **while** $Q_3 \not\subseteq Q_1$
  **do** $Q_1 := Q_1 \cup Q_3$;  $Q_3 := pre(M, A, Q_1) \cap Q_2$ **od**;
  return $Q_1$
**end case**

**function** $pre(M, A, Q)$.

Auxiliary function; returns the exact set of states $Q'$ such that, when the system is in a state $q \in Q'$, agents $A$ can cooperate and enforce the next state to be in $Q$.

return $\{q \mid \exists \alpha_A \forall \alpha_{\mathbb{A}\mathrm{gt} \setminus A}\ o(q, \alpha_A, \alpha_{\mathbb{A}\mathrm{gt} \setminus A}) \in Q\}$

**Fig. 5.3.** The **ATL** model checking algorithm from [Alur et al., 2002]

*Proof.* [Sketch] Each case of the algorithm is called at most $\mathbf{O}(|\varphi|)$ times and terminates after $\mathbf{O}(|\mathfrak{M}|)$ steps [Alur et al., 2002]. The latter is shown by translating the model to a two player game [Alur et al., 2002], and then solving the "invariance game" on it in polynomial time [Beeri, 1980]. Hardness is shown by a reduction of reachability in And-Or-Graphs, which was shown to be **P**-complete (cf. Theorem 4.22, [Immerman, 1981]), to model checking the (constant) $\mathcal{L}_{ATL}$-formula $\langle\!\langle 1 \rangle\!\rangle \Diamond \mathsf{p}$ in a two player game. In each Or-state it is the turn of player 1 and in each And-state it is player 2's turn [Alur et al., 2002]. ■

In the next theorem, we show that the model checking problem of coalition logic is as hard as for **ATL**. To our knowledge, this is a new result; the proof is done by a slight variation of the hardness proof for **ATL** in [Alur et al., 2002] (cf. the proof of Theorem 5.5).

**Theorem 5.6 ($\mathbf{CL_{Ir}}$ and $\mathbf{CL_{IR}}$).** *Model checking $\mathbf{CL_{Ir}}$ and $\mathbf{CL_{IR}}$ is $\mathbf{P}$-complete, and can be done in time $\mathbf{O}(|\mathfrak{M}| \cdot |\varphi|)$, where $|\mathfrak{M}|$ is given by the number of transitions in $\mathfrak{M}$.*

*Proof.* The upper bound follows from the fact that $\mathcal{L}_{CL}$ is a sublanguage of $\mathcal{L}_{ATL}$. We show $\mathbf{P}$-hardness by the following adaption of the reduction of And-Or-Graph reachability from [Alur et al., 2002]. Firstly, we observe that if a state $y$ is reachable from $x$ in graph $G$ then it is also reachable via a path whose length is bounded by the number $n$ of states in the graph. Like in the proof of Theorem 5.5, we take $G$ to be a turn-based CGS in which player 1 "owns" all the Or-states and player 2 "owns" all the And-states. We also label node $y$ with a special proposition $\mathsf{y}$, and replace all the transitions outgoing from $y$ with a deterministic loop. Now, we have that $y$ is reachable from $x$ in $G$ iff $G, x \models \underbrace{\langle\!\langle 1 \rangle\!\rangle \bigcirc \ldots \langle\!\langle 1 \rangle\!\rangle \bigcirc}_{n\text{-times}} \mathsf{y}$. The reduction uses only logarithmic space. ■

It is worth pointing out, however, that checking strategic properties in one-step games is somewhat easier. We recall that $\mathbf{AC^0}$ is the class corresponding to constant depth, unbounded fanin, polynomial size Boolean circuits with AND, OR, and NOT gates [Furst et al., 1984]. We call a formula *flat* if it contains no nested cooperation modalities. Moreover, a formula is *simple* if it is flat and does not include Boolean connectives. For example, the language of "simple $\mathbf{CL}$" consists only of formulae $p$ and $\langle\!\langle A \rangle\!\rangle \bigcirc p$, for $p \in \Pi$ and $A \subseteq \mathbb{A}\mathrm{gt}$.

**Theorem 5.7 (Simple $\mathbf{CL_{Ir}}$ and $\mathbf{CL_{IR}}$ [Laroussinie et al., 2008]).** *Model checking "simple $\mathbf{CL_{Ir}}$" and "simple $\mathbf{CL_{IR}}$" with respect to the number of transitions in the model and the length of the formula is in $\mathbf{AC^0}$.*

*Proof.* [Sketch] For $\mathfrak{M}, q \models \langle\!\langle A \rangle\!\rangle \bigcirc \mathsf{p}$, we construct a 3-level circuit [Laroussinie et al., 2008]. On the first level, we assign one AND gate for each possible coalition $B$ and $B$'s collective choice $\alpha_B$. The output of the gate is "true" iff $\alpha_B$ leads to a state satisfying $\mathsf{p}$ for each response of $\mathbb{A}\mathrm{gt} \setminus B$. On the second level, there is one OR gate per possible coalition $B$ that connects all $B$'s gates from the first level and outputs "true" iff there is any successful strategy for $B$. On the third level, there is a single AND gate that selects the right output (i.e., the one for coalition $A$). ■

### 5.3.2 ATL and CL: Imperfect Information

In contrast to the perfect information setting, analogous fixedpoint characterisations do *not* hold for the incomplete information semantics over $\mathcal{L}_{ATL}$ because the choice of a particular action at a state $q$ has non-local consequences: It automatically fixes choices at all states $q'$ indistinguishable from $q$

for the coalition $A$. Moreover, the agents' ability to *identify* a strategy as winning also varies throughout the game in an arbitrary way (agents can learn as well as forget). This suggests that winning strategies cannot be synthesised incrementally. In order to check $\mathfrak{M}, q \models \langle\langle A \rangle\rangle \gamma$ (where $\gamma$ includes no nested cooperation modalities), the following procedure suffices. Firstly, we guess a uniform strategy $s_A$ of team $A$ (by calling an **NP** oracle), and then we verify the strategy by pruning $\mathfrak{M}$ accordingly (removing all the transitions that are not going to be executed according to $s_A$) and model checking the $\mathcal{L}_{CTL}$-formula $\mathsf{A}\gamma$ in the resulting model. For nested cooperation modalities, we proceed recursively (bottom up). Since model checking **CTL** can be done in polynomial deterministic time, the procedure runs in polynomial deterministic time with calls to an **NP** oracle, which demonstrates the inclusion in $\mathbf{\Delta_2^P} = \mathbf{P^{NP}}$ [Schobbens, 2004]. As it turns out, a more efficient procedure does not exist, which is confirmed by the following result.

**Theorem 5.8 ($\mathbf{ATL_{ir}}$ [Schobbens, 2004; Jamroga and Dix, 2008]).** *Model checking* $\mathbf{ATL_{ir}}$ *is* $\mathbf{\Delta_2^P}$*-complete in the number of transitions in the model and the length of the formula.*

*Proof.* [Sketch] The discussion above proves membership in $\mathbf{\Delta_2^P}$. $\mathbf{\Delta_2^P}$-hardness is shown in [Jamroga and Dix, 2008] through a reduction of *sequential satisfiability* ($\text{SNSAT}_2$), a standard $\mathbf{\Delta_2^P}$-complete problem (cf. Definition 4.15, [Laroussinie et al., 2001]). The idea is that there are two agents where one agent tries to verify a (nested) propositional formula and a second agent tries to refute it. A winning strategy of the "verifier agent" corresponds to a satisfying valuation of the formula. Uniformity of the verifier's strategy is needed to ensure that identical proposition symbols, occurring at different places in the formula, are assigned the same truth values. ■

Now we consider the incomplete information setting for coalition logic. It is easy to see that the *iR*- and *ir*-semantics are equivalent for $\mathcal{L}_{CL}$ since $\bigcirc$ is the only temporal operator, and thus only the first action in a strategy matters. As a consequence, whenever there is a successful *iR*-strategy for agents $A$ to enforce $\bigcirc\varphi$, then there is also an *ir*-strategy for $A$ to obtain the same. Perfect recall of the history does not matter in one-step games.

**Theorem 5.9 ($\mathbf{CL_{ir}}$ and $\mathbf{CL_{iR}}$).** *Model checking* $\mathbf{CL_{ir}}$ *and* $\mathbf{CL_{iR}}$ *is* $\mathbf{P}$*-complete wrt the number of transitions in the model and the length of the formula, and can be done in time* $\mathbf{O}(|\mathfrak{M}| \cdot |\varphi|)$.

*Proof.* The **P**-hardness follows from Theorem 5.6 (perfect information CGSs can be seen as a special kind of ICGSs where the indistinguishability relations contain only the reflexive loops). For the upper bound, we use the following algorithm. For $\mathfrak{M}, q \models \langle\langle A \rangle\rangle \bigcirc \mathsf{p}$, we check if there is a collective action $\alpha_A$ such that for all responses $\alpha_{\mathbb{A}\text{gt}\setminus A}$ we have that $\bigcup_{\{q' | q \sim_A q'\}} \{o(q', \alpha_A, \alpha_{\mathbb{A}\text{gt}\setminus A})\} \subseteq \pi(\mathsf{p})$. For $\langle\!\langle A \rangle\!\rangle \bigcirc \varphi$ with nested cooperation modalities, we proceed recursively (bottom up). ■

**Theorem 5.10 (Simple $\mathsf{CL}_{\mathsf{ir}}$ and $\mathsf{CL}_{\mathsf{iR}}$).** *Model checking "simple" formulae of $\mathsf{CL}_{\mathsf{ir}}$ and $\mathsf{CL}_{\mathsf{iR}}$ with respect to the number of transitions in the model and the length of the formula is in* $\mathbf{AC^0}$.

*Proof.* For $\mathfrak{M}, q \models \langle\!\langle A \rangle\!\rangle \bigcirc \mathsf{p}$, we extend the procedure from [Laroussinie et al., 2008] by creating one copy of the circuit per $q' \in \text{img}(q, \sim_A)$. Then, we add a single AND gate on the fourth level of the circuit, that takes the output of those copies and returns "true" iff $A$ have a strategy that is successful from all states indistinguishable from $q$. ■

That leaves us with the issue of $\mathcal{L}_{ATL}$ with the semantics assuming imperfect information and perfect recall. To our knowledge, there is no formal proof published in the literature regarding the complexity of model checking $\mathcal{L}_{ATL}$ with *iR*-strategies. However, the problem is commonly believed to be undecidable and just recently a proof has been proposed by Dima and Tiplea (June 2010).

*Conjecture 5.11 ($\mathsf{ATL}_{\mathsf{iR}}$ [Alur et al., 2002]).* Model checking $\mathsf{ATL}_{\mathsf{iR}}$ is undecidable.

### 5.3.3 ATL*

We now turn to model checking logics over broader subsets of $\mathcal{L}_{ATL^*}$. In the first case we consider perfect recall strategies in the perfect information setting. The complexity results established here are based on an automata-theoretic approach which is explained below.

Let $\mathfrak{M}$ be a CGS and $\langle\!\langle A \rangle\!\rangle \psi$ be an $\mathcal{L}_{ATL^*}$-formula (where we assume that $\psi$ is an $\mathcal{L}_{LTL}$-formula). Given a strategy $s_A$ of $A$ and a state $q$ in $\mathfrak{M}$ the model can be unfolded into a $q$-rooted tree representing all possible behaviours with agents $A$ following their strategy $s_A$. This structure can be seen as the tree *induced* by $out(q, s_A)$ and we will refer to it as a $(q, A)$-*execution tree.* Note that every strategy profile for $A$ may result in a different execution tree. Now, a Büchi tree automaton $\mathcal{A}_{\mathfrak{M},q,A}$ can be constructed that accepts exactly the $(q, A)$-execution trees [Alur et al., 2002].

Secondly, it was shown that one can construct a Rabin tree automaton which accepts all trees that satisfy the $\mathcal{L}_{CTL^*}$-formula $\mathsf{A}\psi$ [Emerson and Sistla, 1984]. Hence, the $\mathcal{L}_{ATL^*}$-formula $\langle\!\langle A \rangle\!\rangle \psi$ is satisfied in $\mathfrak{M}, q$ if there is a tree accepted by $\mathcal{A}_{\mathfrak{M},q,A}$ (i.e., it is a $(q, A)$-execution tree) and by $\mathcal{A}_{\psi}$ (i.e., it is a model of $\mathsf{A}\psi$).

**Theorem 5.12 ($\mathbf{ATL}^*_{\mathbf{IR}}$ [Alur et al., 2002]).** *Model checking $\mathbf{ATL}^*_{\mathbf{IR}}$ is* **2EXPTIME**-*complete in the number of transitions in the model and the length of the formula.*

*Proof.* [Sketch] We briefly analyse the complexity for the procedure described above. Firstly, the Büchi tree automaton $\mathcal{A}_{\mathfrak{M},q,A}$ is built by considering the states $A$ is effective for [Alur et al., 2002]. That is, in a state of the automaton corresponding to a state $q \in Q$ of $\mathfrak{M}$ the automaton nondeterministically chooses a sequence $(q'_1, q'_2, \dots, q'_n)$ of successors of $q$ such that $A$ has a common action to guarantee that the system will end up in one of the states $\{q'_1, q'_2, \dots, q'_n\}$ in the next step. It is assumed that the sequence is minimal. Incrementally, this models any $s_A$ strategy of $A$ and thus accepts all $(q, A)$-execution trees. The transition function of the automaton is constructed in the described way. As the number of transitions in each state of the automaton is bounded by the move combinations of agents $A$, the size of the automaton, $|\mathcal{A}_{\mathfrak{M},q,A}|$, is bounded by $\mathbf{O}(|\mathfrak{M}|)$. All states are defined as acceptance states, such that $\mathcal{A}_{\mathfrak{M},q,A}$ accepts all possible execution trees of $A$.

Following the construction of [Emerson and Sistla, 1984], the automaton $\mathcal{A}_\psi$ is a Rabin tree automaton with $2^{2^{\mathbf{O}(|\psi|)}}$ states and $2^{\mathbf{O}(|\psi|)}$ Rabin pairs.

The product automaton $\mathcal{A}_\psi \times \mathcal{A}_{\mathfrak{M},q,A}$, accepting the trees accepted by both automata, is a Rabin tree automaton with $n := \mathbf{O}(|\mathcal{A}_\psi| \cdot |\mathcal{A}_{\mathfrak{M},q,A}|)$ many states and $r := 2^{\mathbf{O}(|\psi|)}$ many Rabin pairs (note that $\mathcal{A}_{\mathfrak{M},q,A}$ can be seen as a Rabin tree automaton with one Rabin pair composed of the states of the automaton and the empty set). Finally, to determine whether the language accepted by the product automaton is empty can be done in time $\mathbf{O}(n \cdot r)^{3r}$ [Emerson and Jutla, 1988; Pnueli and Rosner, 1989]; hence, the algorithm runs in time $|\mathfrak{M}|^{2^{\mathbf{O}(|\psi|)}}$ (it might be employed at each state of the model and for each subformula).

The lower bound is shown by a reduction of the **2EXPTIME**-complete problem of the realisability of **LTL**-formulae [Pnueli and Rosner, 1989; Rosner, 1992; Alur et al., 2002] (cf. Theorem 4.24). ■

The next result shows that model checking $\mathcal{L}_{ATL^*}$ *with memoryless strategies* is no worse than for **LTL** and **CTL**$^\star$ for both perfect and imperfect information.

**Theorem 5.13 ($\mathbf{ATL}^*_{\mathbf{ir}}$ and $\mathbf{ATL}^*_{\mathbf{Ir}}$ [Schobbens, 2004]).** *Model checking* $\mathbf{ATL}^*_{\mathbf{ir}}$ *and* $\mathbf{ATL}^*_{\mathbf{Ir}}$ *is* **PSPACE**-*complete in the number of transitions in the model and the length of the formula.*

*Proof.* [Sketch] $\mathcal{L}_{LTL}$ is contained in $\mathcal{L}_{ATL^*}$ which makes $\mathcal{L}_{ATL^*}$ under the perfect information memoryless semantics to be at least **PSPACE**-hard.

On the other hand, there is a **PSPACE** algorithm for model checking $\mathcal{L}_{ATL^*}$ under the imperfect information memoryless semantics. Consider the formula $\langle\langle A\rangle\rangle\psi$ where $\psi$ is an $\mathcal{L}_{LTL}$-formula. Then, an *ir*-strategy $s_A$ for $A$ is guessed and the model is "trimmed" according to $s_A$, i.e. all transitions which cannot occur by following $s_A$ are removed. A memoryless strategy can be guessed in polynomially many steps, and hence also using only polynomially many memory cells. In the new model the $\mathcal{L}_{CTL^*}$-formula $\mathsf{A}\psi$ is checked. This procedure can be performed in $\mathbf{NP}^{\mathbf{PSPACE}}$, which renders the complexity of the whole language to be in $\mathbf{P}^{\mathbf{NP}^{\mathbf{PSPACE}}} = \mathbf{PSPACE}$. ■

### 5.3.4 ATL$^+$ and EATL$^+$

We consider the more limited language $\mathcal{L}_{ATL^+}$. Boolean combinations of path formulae prevent us from using the fixed-point characterisations for model checking. Instead, given a formula $\langle\langle A\rangle\rangle\psi$ with no nested cooperation modalities, we can guess a (memoryless) strategy of $A$, "trim" the model accordingly, and model check the $\mathcal{L}_{CTL^+}$-formula $\mathsf{A}\psi$ in the resulting model. Since the model checking problem for **CTL**$^+$ is $\mathbf{\Delta_2^P}$-complete (cf. Theorem 5.4), we get that the overall procedure runs in time $\mathbf{\Delta_2^{P^{\Delta_2^P}}} = \mathbf{\Delta_3^P}$ [Schobbens, 2004].

**Theorem 5.14 (ATL$_{ir}^+$ and ATL$_{Ir}^+$ [Schobbens, 2004]).** *Model checking* **ATL**$_{ir}^+$ *and* **ATL**$_{Ir}^+$ *is* $\mathbf{\Delta_3^P}$*-complete in the number of transitions in the model and the length of the formula.*

*Proof.* [Sketch] The above procedure shows the membership. In the incomplete information case one has to guess a *uniform* strategy. Again, it is essential that a strategy can be guessed in *polynomially many steps*, which is indeed the case for *Ir*- and *ir*-strategies. The hardness proof can be obtained by a reduction of the standard $\mathbf{\Delta_3^P}$-complete problem SNSAT$_3$, cf. [Schobbens, 2004] for the construction. ■

What about **ATL**$_{IR}^+$? It has been believed that verification with $\mathcal{L}_{ATL^+}$ is $\mathbf{\Delta_3^P}$-complete for perfect recall strategies, too. However, it turns out that the complexity of **ATL**$_{IR}^+$ model checking is apparently much harder, namely **PSPACE** [Bulling and Jamroga, 2010a,b]. Since the $\mathbf{\Delta_3^P}$-completeness for memoryless semantics *is* correct, we get that memory makes verification harder already for $\mathcal{L}_{ATL^+}$, and not just for $\mathcal{L}_{ATL^*}$ as it was believed before. We treat this case in more detail in Section 9.1; here, we just present the result.

**Theorem 5.15 (ATL$_{IR}^+$ [Bulling and Jamroga, 2010b]).** *Model checking* **ATL**$_{IR}^+$ *is* **PSPACE***-complete with respect to the number of transitions in the model and the length of the formula. It is* **PSPACE***-complete even for turn-based models with two agents and "flat"* $\mathcal{L}_{ATL^+}$*-formulae.*

| | *Ir* | *IR* | *ir* | *iR* |
|---|---|---|---|---|
| Simple $\mathcal{L}_{CL}$ | $\mathbf{AC^0}$ | $\mathbf{AC^0}$ | $\mathbf{AC^0}$ | $\mathbf{AC^0}$ |
| $\mathcal{L}_{CL}$ | **P** | **P** | **P** | **P** |
| $\mathcal{L}_{ATL}$ | **P** | **P** | $\mathbf{\Delta_2^P}$ | Undecidable† |
| $\mathcal{L}_{ATL^+}$ | $\mathbf{\Delta_3^P}$ | **PSPACE** | $\mathbf{\Delta_3^P}$ | Undecidable† |
| $\mathcal{L}_{ATL^*}$ | **PSPACE** | **2EXPTIME** | **PSPACE** | Undecidable† |

**Fig. 5.4.** Overview of the model checking complexity results for explicit models. All results except for "Simple CL" are completeness results. Each cell represents the logic over the language given in the row using the semantics given in the column. † These problems are believed to be undecidable, though no formal proof has been proposed yet (cf. Conjectures 5.11, 5.16, and 5.17).

The following conjectures are immediate consequences of Conjecture 5.11 as $\mathcal{L}_{ATL}$ is a fragment of $\mathcal{L}_{ATL^*}$ as well as $\mathcal{L}_{ATL^+}$.

*Conjecture 5.16 (*$\mathbf{ATL^*_{iR}}$*).* Model checking $\mathbf{ATL^*_{iR}}$ is undecidable.

*Conjecture 5.17 (*$\mathbf{ATL^+_{iR}}$*).* Model checking $\mathbf{ATL^+_{iR}}$ is undecidable.

Figure 5.4 presents an overview of the model checking complexity results for explicit models.

## 5.4 Model Checking on Implicit Models

In the following we consider how the model checking results change if the size of the models is not given in the most straightforward way (i.e. in terms of the number of states and transitions) but rather in a compressed form.

### 5.4.1 Implicit Models

We have seen several complexity results for the model checking problem in logics like **LTL**, **CTL**, and **ATL**. Some of these results are quite attractive: One usually cannot hope to achieve verification with complexity better than *linear*.

However, it is important to remember that these results measure the complexity with respect to the *size of the underlying model*. Often, these models are so big, that an *explicit* representation is not possible and we have to represent the model in a "compressed" way. To give a simple illustration, consider the famed primality problem: checking whether a given natural number $n$ is prime. The well-known algorithm uses $\sqrt{n}$-many divisions and thus runs in polynomial time *when the input is represented in unary*. But a symbolic representation of $n$ needs only $\log(n)$ bits and thus the above algorithm runs in *exponential* time with respect to its size. This does not necessarily imply that the problem itself is of exponential complexity. In fact, the famous and deep result of Agrawal, Kayal and Saxena shows that the primality problem *can* be solved in polynomial time.

One may consider model checking of temporal and strategic logics for such highly compressed representations (in terms of *state space compression* and *modularisation*). However, such a rigorous compressed representation is not the only way in which the model checking complexity can be influenced. Another important factor is how we encode the transition function. So far, we assumed that the size of a model is measured with respect to the number of transitions in the model.

In this section we consider the complexity of the model checking problem *with respect to the number of states, agents, and an implicitly encoded transition function* rather than the (explicit) number of transitions. It is easy to see that, for CGSs, the number of transitions can be exponential in the number of states and agents. Therefore, all the algorithms presented in Section 10.1 give us only exponential time bounds provided that the transition function is encoded sufficiently small.

*Remark 5.18 ([Alur et al., 2002; Jamroga and Dix, 2005]).* Let $n$ be the number of states in a concurrent game structure $\mathfrak{M}$, let $k$ denote the number of agents, and $d$ the maximal number of available decisions (moves) per agent per state. Then, $m = \mathbf{O}(nd^k)$. Therefore the **ATL$_{\mathbf{IR}}$** model checking algorithm from [Alur et al., 2002] runs in time $\mathbf{O}(nd^k l)$, and hence its complexity is exponential if the number of agents is a parameter of the problem.

In comparison, for an *unlabelled* transition system with $n$ states and $m$ transitions, we have that $m = \mathbf{O}(n^2)$. This means that **CTL** model checking is in **P** also with respect to the number of states in the model and the length of the formula. The following theorem is an immediate corollary of this fact (and Theorem 5.1).

**Theorem 5.19 ([Clarke et al., 1986]). CTL** *model checking over unlabelled transition systems is* **P***-complete in the number of states and the length of the formula, and can be done in time* $\mathbf{O}(n^2 l)$.

For **ATL** and concurrent game structures, however, the situation is different. In the following we make precise what we mean by a compressed transition function.

*Implicit concurrent game structures* (called this way first in [Laroussinie et al., 2006], but already present in the ISPL modelling language behind MCMAS [Raimondi and Lomuscio, 2004; Raimondi, 2006]) are defined similarly to a CGS but the transition function is encoded in a particular way often allowing for a more compact representation than the explicit transition table. Formally, an *implicit* CGS is given by $\mathfrak{M} = \langle \mathbb{A}\text{gt}, Q, \Pi, \pi, Act, d, \hat{o}\rangle$ where $\hat{o}$, the *encoded transition function*, is given by a sequence

$$((\varphi_0^r, q_0^r), \ldots, (\varphi_{t_r}^r, q_{t_r}^r))_{r=1,\ldots,|Q|}$$

where $t_r \in \mathbb{N}_0$, $q_i^r \in Q$ and each $\varphi_i^r$ is a Boolean combination of propositions $\mathsf{exec}_\alpha^{\mathsf{j}}$ where $j \in \mathbb{A}\text{gt}$, $\alpha \in Act$, $i = 1, \ldots, t$ and $r = 1, \ldots, |Q|$. It is required that $\varphi_{t_r}^r = \top$. The term $\mathsf{exec}_\alpha^{\mathsf{j}}$ stands for "agent $j$ executes action $\alpha$". We use $\varphi[\alpha_1, \ldots, \alpha_k]$ to refer to the Boolean formula over $\{\top, \bot\}$ obtained by replacing $\mathsf{exec}_\alpha^{\mathsf{a_j}}$ with $\top$ (resp. $\bot$) if $\alpha_j = \alpha$ (resp. $\alpha_j \neq \alpha$). The encoded transition function induces a standard transition function $o_{\hat{o}}$ as follows:

$$o_{\hat{o}}(q_i, \alpha_1, \ldots, \alpha_k) = q_j^i \text{ where } j = \min\{\kappa \mid \varphi_\kappa^i[\alpha_1, \ldots, \alpha_k] \equiv \top\}.$$

That is, $o_{\hat{o}}(q_i, \alpha_1, \ldots, \alpha_k)$ returns the state belonging to the formula $\varphi_\kappa^i$ (associated with state $q_i$) with the minimal index $\kappa$ that evaluates to "true" given the actions $\alpha_1, \ldots, \alpha_k$. We use $\hat{o}(q_i, \alpha_1, \ldots, \alpha_k)$ to refer to $o_{\hat{o}}(q_i, \alpha_1, \ldots, \alpha_k)$. Note that the function is well defined as the last formula in each sequence is given by $\top$: No deadlock can occur. The size of $\hat{o}$ is defined as $|\hat{o}| = \sum_{r=1,\ldots,|Q|} \sum_{j=1,\ldots,t_r} |\varphi_j^r|$, that is, the sum of the sizes of all formulae. Hence, the size of an implicit CGS is given by $|Q| + |\mathbb{A}\text{gt}| + |\hat{o}|$. We recall, that the size of an explicit CGS is $|Q| + |\mathbb{A}\text{gt}| + m$ where $m$ is the number of transitions. Finally, we require that the encoding of the transition function is reasonably compact, that is, $|\hat{o}| \leq \mathbf{O}(|o_{\hat{o}}|)$.

Now, why should the model checking complexity change for implicit CGSs? Firstly, one can observe that we can take the trivial encoding of an explicit transition function yielding an implicit CGS that has the same size as the explicit CGS. This implies that all the *lower bounds* proven before are still valid.

**Proposition 5.20 ([Bulling et al., 2010]).** *Model checking with respect to implicit* CGSs *is at least as hard as model checking over explicit* CGSs *for* $\mathbf{CTL}$, $\mathbf{CTL}^\star$,$\mathbf{LTL}$, $\mathbf{ATL_{xy}}$, $\mathbf{ATL^*_{xy}}$, *and* $\mathbf{ATL^+_{xy}}$ *for* $x \in \{i, I\}$ *and* $y \in \{r, R\}$.

Therefore, we focus on the question whether model checking can become more difficult for implicit CGSs. Unfortunately, the answer is *yes*: Model checking can indeed become more difficult.

We illustrate this by considering the presented algorithm for solving the $\mathbf{ATL_{IR}}$ model checking problem. It traverses all transitions and as transitions are considered explicitly in the input, the algorithm runs in polynomial time. But if we choose an encoding $\hat{o}$ that is significantly smaller than the explicit number of transitions, the algorithm still has to check all transitions, yet now the number of transitions can be *exponential* with respect to the input of size $|Q| + |\mathbb{A}\mathrm{gt}| + |\hat{o}|$.

Henceforth, we are interested in the cases in which the size of the encoded transition function is much smaller, in particular, when the size of the encoding is polynomial with respect to the *number of states and agents*. This is the reason why we will often write that we measure the input in terms of states ($n$) and agents ($k$), neglecting the size of $\hat{o}$ when it is supposed to be polynomial in $n, k$.

*Remark 5.21.* An alternative view is to assume that the transition function is provided by an external procedure (a "black box") that runs in polynomial time, similar to an oracle [Jamroga and Dix, 2005]. In our opinion, this view comes along with some technical disadvantages, and we will not discuss it here.

### 5.4.2 ATL and CL

As argued above the complexity of $\mathbf{O}(ml)$ may (but does not have to) include potential intractability if the transition function is represented more succinctly. The following result supports this observation.

**Theorem 5.22 ([Laroussinie et al., 2008; Jamroga and Dix, 2005, 2008]).** *Model checking* $\mathsf{ATL_{IR}}$ *and* $\mathsf{ATL_{Ir}}$ *over implicit* CGSs *is* $\mathbf{\Delta_3^P}$*-complete with respect to the size of the model and the length of the formula (l).*

*Proof.* [Sketch] The idea of the proof for the lower bound is clear if we reformulate the model checking of $\mathfrak{M}, q \models \langle\langle a_1, \ldots, a_r \rangle\rangle \bigcirc \varphi$ as

$$\exists(\alpha_1, \ldots, \alpha_r)\forall(\alpha_{r+1}, \ldots, \alpha_k)\ M, o(q, \alpha_1, \ldots, \alpha_k) \models \varphi,$$

which closely resembles QSAT $_2$, a typical $\mathbf{\Sigma_2^P}$-complete problem. A reduction of this problem to our model checking problem is straightforward: For each instance of QSAT $_2$, we create a model where the values of propositional variables $p_1, \ldots, p_r$ are "declared" by agents $A$ and the values of $p_{r+1}, \ldots, p_k$ by $\mathbb{A}\mathrm{gt} \setminus A$. The subsequent transition leads to a state labeled by proposition $\mathsf{yes}$ iff the given Boolean formula holds for the underlying valuation of $p_1, \ldots, p_k$. Then, QSAT $_2$ reduces to model checking formula $\langle\langle a_1, \ldots, a_r \rangle\rangle \bigcirc \mathsf{yes}$ [Jamroga and Dix, 2005]. In order to obtain $\mathbf{\Delta_3^P}$-hardness, the above schema is combined with nested cooperation modalities, which yields a rather technical reduction of the SNSAT$_3$ problem that can be found in [Laroussinie et al., 2008].

For the upper bound, we consider the following algorithm for checking $\mathfrak{M}, q \models \langle\langle A \rangle\rangle\gamma$ with no nested cooperation modalities. Firstly, guess a strategy $s_A$ of the proponents and fix $A$'s actions to the ones described by $s_A$. Then check if $\mathsf{A}\gamma$ is true in state $q$ of the resulting model by asking an oracle about the existence of a counterstrategy $s_{\bar{A}}$ for $\mathbb{A}\mathrm{gt} \setminus A$ that falsifies $\gamma$ and reverting the oracle's answer. The evaluation takes place by calculating $\hat{o}$ (which takes polynomially many steps) regarding the actions prescribed by $(s_A, s_{\bar{A}})$ at most $|Q|$ times. For nested cooperation modalities, we proceed recursively (bottom-up). ■

Surprisingly, the imperfect information variant of **ATL** is no harder than the perfect information one under this measure:

**Theorem 5.23 ([Jamroga and Dix, 2008]).** *Model checking* $\mathbf{ATL_{ir}}$ *over implicit* CGSs *is* $\mathbf{\Delta_3^P}$*-complete with respect to the size of the model and the length of the formula. This is the same complexity as for model checking* $\mathbf{ATL_{Ir}}$ *and* $\mathbf{ATL_{IR}}$*.*

*Proof.* [Sketch] For the upper bound, we use the same algorithm as in checking $\mathbf{ATL_{Ir}}$. For the lower bound, we observe that $\mathbf{ATL_{Ir}}$ can be embedded in $\mathbf{ATL_{ir}}$ by explicitly assuming perfect information of agents (through the minimal reflexive indistinguishability relations). ■

The $\mathbf{\Delta_3^P}$-hardness proof in Theorem 5.22 uses the "next time" and "until" temporal operators in the construction of an **ATL** formula that simulates $\mathrm{SNSAT}_3$ [Laroussinie et al., 2008]. However, the proof can be modified so that only the "next time" sublanguage of $\mathcal{L}_{ATL}$ is used. We obtain thus an analogous result for coalition logic.

**Theorem 5.24 ([Bulling, 2010]).** *Model checking* $\mathbf{CL_{IR}}$*,* $\mathbf{CL_{Ir}}$*,* $\mathbf{CL_{ir}}$*, and* $\mathbf{CL_{iR}}$ *over implicit* CGSs *is* $\mathbf{\Delta_3^P}$*-complete with respect to the size of the model and the length of the formula. Moreover, it is* $\mathbf{\Sigma_2^P}$*-complete for the "simple" variants of* CL*.*

The proof and details of the new construction can be found on page 322.

It is worth mentioning that model checking "Positive **ATL**" (i.e., the fragment of $\mathcal{L}_{ATL}$ where negation is allowed only on the level of literals) is $\mathbf{\Sigma_2^P}$-complete with respect to the size of implicit CGSs, and the length of formulae for the *IR*, *Ir*, and *ir*-semantics [Jamroga and Dix, 2008]. The same applies to "Positive CL", the analogous variant of coalition logic.

### 5.4.3 CTL and CTL$^+$ Revisited

At the beginning of Section 5.4.1, we have mentioned that the complexity of model checking computation tree logic is still polynomial even if we measure the size of models with the number of states rather than transitions. That is certainly true for unlabelled transition systems (i.e., the original models of **CTL**). For concurrent game structures, however, this is no longer the case.

**Theorem 5.25 ([Bulling et al., 2010]).** *Model checking* **CTL** *over implicit* CGSs *is* $\mathbf{\Delta_2^P}$*-complete with respect to the size of the model and the length of the formula.*

*Proof.* [Sketch] For the upper bound, we observe that $\mathfrak{M}, q \models^{\mathsf{CTL}} \mathsf{E}\gamma$ iff $\mathfrak{M}, q \models_{IR} \langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\gamma$ which is in turn equivalent to $\mathfrak{M}, q \models_{Ir} \langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\gamma$. In other words, $\mathsf{E}\gamma$ holds iff the grand coalition has a *memoryless* strategy to achieve $\gamma$. Thus, we can verify $\mathfrak{M}, q \models \mathsf{E}\gamma$ (with no nested path quantifiers) as follows: we guess a strategy $s_{\mathbb{A}\mathrm{gt}}$ for $\mathbb{A}\mathrm{gt}$ (in polynomially many steps), then we construct the resulting model $\mathfrak{M}'$ by asking $\hat{o}$ which transitions are enabled by following the strategy $s_A$ and check if $\mathfrak{M}', q \models \mathsf{E}\gamma$ and return the answer. Note that $\mathfrak{M}'$ is an *unlabelled transition system*, so constructing $\mathfrak{M}'$ and checking $\mathfrak{M}', q \models \mathsf{E}\gamma$ can be done in polynomial time. For nested modalities, we proceed recursively.

For the lower bound, we sketch the reduction of the satisfiability problem (SAT) to model checking $\mathcal{L}_{CTL}$-formulae with only one path quantifier. For propositional variables $p_1, \ldots, p_k$ and boolean formula $\varphi$, we construct an implicit CGS where the values of $p_1, \ldots, p_k$ are "declared" by agents $\mathbb{A}\mathrm{gt} = \{a_1, \ldots, a_k\}$ (in parallel). The subsequent transition leads to a state labeled by proposition $\mathsf{yes}$ iff $\varphi$ holds for the underlying valuation of $p_1, \ldots, p_k$. Then, SAT reduces to model checking formula $\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle \bigcirc \mathsf{yes}$. The reduction of $\mathrm{SNSAT}_2$ (to model checking $\mathcal{L}_{CTL}$-formulae with nested path quantifiers) is an extension of the SAT reduction, analogous to the one in [Jamroga and Dix, 2006, 2008]. ■

It turns out that the complexity of $\mathbf{CTL}^+$ does not increase when we change the models to implicit concurrent game structures: It is still $\mathbf{\Delta_2^P}$.

**Theorem 5.26 ([Bulling et al., 2010]).** *Model checking* $\mathbf{CTL}^+$ *over implicit* CGSs *is* $\mathbf{\Delta_2^P}$*-complete with respect to the size of the model and the length of the formula.*

*Proof.* [Sketch] The lower bound follows from Theorem 5.4 and Proposition 5.20.

For the upper bound, we observe that the $\mathbf{CTL}^+$ model checking algorithm in [Laroussinie et al., 2001] verifies $\mathfrak{M}, q \models \mathsf{E}\gamma$ by guessing a finite history $h$ with length $|Q_M| \cdot |\gamma|$, and then checking $\gamma$ on $h$. We recall that $\mathsf{E}\gamma \equiv \langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\gamma$. Thus, for a concurrent game structure, each transition in $h$ can be determined by guessing an action profile in $O(|\mathbb{A}\mathrm{gt}|)$ steps, calculating $\hat{o}$ wrt the guessed profile, and the final verification whether $\gamma$ holds on the *finite* sequence $h$ which can be done in deterministic polynomial time (cf. [Bulling and Jamroga, 2010a]). Consequently, we can implement this procedure by a nondeterministic Turing machine that runs in polynomial time. For nested path quantifiers, we proceed recursively which shows that the model checking problem can be solved by a polynomial time Turing machine with calls to an **NP**-oracle. ■

We will use the last result in the analysis of $\mathsf{ATL}^+$ in Section 5.4.4.

### 5.4.4 $\mathsf{ATL}^*$, $\mathsf{ATL}^+$, and $\mathsf{EATL}^+$

**Theorem 5.27.** *Model checking* $\mathsf{ATL}^*_{\mathsf{Ir}}$ *and* $\mathsf{ATL}^*_{\mathsf{ir}}$ *over implicit* CGSs *is* **PSPACE**-*complete with respect to the size of the model and the length of the formula.*

*Proof.* The lower bound follows from Theorem 5.13 and Proposition 5.20.

For the upper bound, we model check $\mathfrak{M}, q \models \langle\langle A \rangle\rangle \gamma$ by guessing a memoryless strategy $s_A$ for coalition $A$, then we guess a counterstrategy $s_{\bar{A}}$ of the opponents. Having a complete strategy profile, we proceed as in the proof of Theorem 5.25 and check the $\mathsf{LTL}$ path formula $\gamma$ on the resulting (polynomial model) $\mathfrak{M}'$ which can be done in polynomial space (Theorem 5.13). For nested cooperation modalities, we proceed recursively. ■

**Theorem 5.28 ([Laroussinie et al., 2008]).** *Model checking* $\mathsf{ATL}^*_{\mathsf{IR}}$ *over implicit* CGSs *is* **2EXPTIME**-*complete with respect to the size of the model and the length of the formula.*

*Proof.* The lower bound follows from Theorem 5.12 and Proposition 5.20. For the upper bound, we have to modify the algorithm given in the proof of Theorem 5.12 such that it is capable of dealing with implicit models. More precisely, we need to modify the construction of the Büchi automaton $\mathcal{A}_{\mathfrak{M},q,A}$ that is used to accept the $(q, A)$-execution trees. Before, we simply checked all the moves of $A$ in polynomial time and calculated the set of states $A$ is effective for (as the moves are bounded by the number of transitions). Here, we have to incrementally generate all these moves from $A$ using $\hat{o}$. This may take exponential time (as there can be exponentially many moves in terms of the number of states and agents). However, as this can be done independently of the non-emptiness check, the overall runtime of the algorithm is still double exponential. ■

**Theorem 5.29 ([Laroussinie et al., 2008]).** *Model checking* $\mathsf{ATL}^+_{\mathsf{Ir}}$, *and* $\mathsf{ATL}^+_{\mathsf{ir}}$ *over implicit* CGSs *is* $\mathbf{\Delta_3^P}$-*complete with respect to the size of the model and the length of the formula.*

*Proof.* The lower bounds follow from Theorem 5.14 and Proposition 5.20. For the upper bound we model check $\mathfrak{M}, q \models \langle\langle A \rangle\rangle \gamma$ by guessing a memoryless strategy $s_A$ for coalition $A$, and constructing an *unlabelled transition*

| | *Ir* | *IR* | *ir* | *iR* |
|---|---|---|---|---|
| Simple $\mathcal{L}_{CL}$ | $\mathbf{\Sigma_2^P}$ | $\mathbf{\Sigma_2^P}$ | $\mathbf{\Sigma_2^P}$ | $\mathbf{\Sigma_2^P}$ |
| $\mathcal{L}_{CL}$ | $\mathbf{\Delta_3^P}$ | $\mathbf{\Delta_3^P}$ | $\mathbf{\Delta_3^P}$ | $\mathbf{\Delta_3^P}$ |
| $\mathcal{L}_{ATL}$ | $\mathbf{\Delta_3^P}$ | $\mathbf{\Delta_3^P}$ | $\mathbf{\Delta_3^P}$ | Undecidable$^\dagger$ |
| $\mathcal{L}_{ATL^+}$ | $\mathbf{\Delta_3^P}$ | **PSPACE** | $\mathbf{\Delta_3^P}$ | Undecidable$^\dagger$ |
| $\mathcal{L}_{ATL^*}$ | **PSPACE** | **2EXPTIME** | **PSPACE** | Undecidable$^\dagger$ |

**Fig. 5.5.** Overview of the model checking complexity results for implicit CGS. All results are completeness results. Each cell represents the logic over the language given in the row using the semantics given in the column. $^\dagger$ These problems are believed to be undecidable, though no formal proof has been proposed yet.

*system* $\mathfrak{M}'$ as follows. For each state $q_i$ we evaluate formulae contained in $((\varphi_0^i, q_0^i), \ldots, (\varphi_{t_i}^i, q_{t_i}^i))$ according to the guessed strategy. Then, we introduce a transition from $q_i$ to $q_j^i$ if $(\bigwedge_{k=0,\ldots,j-1} \neg\varphi_k^i) \wedge \varphi_j^i$ is satisfiable (i.e., there is a countermove of the opponents such that $\varphi_i^j$ is true and $j$ is the minimal index). This is the case iff the opponents have a strategy to enforce the next state to be $q_j^i$. These polynomially many tests can be done by *independent* calls of an **NP**-oracle. The resulting model $\mathfrak{M}'$ is an explicit CGS of polynomial size regarding the number of states and agents. Finally, we apply $\mathbf{CTL}^+$ model checking to $\mathsf{A}\gamma$ which can be done in time $\mathbf{\Delta_2^P}$. ■

Finally, we consider the case for perfect recall strategies. The lower and upper bound directly follow from the proof of Theorem 5.15 (also cf. Section 9.3).

**Theorem 5.30 ([Bulling and Jamroga, 2010a]).** *Model checking* $\mathbf{ATL}^+_{\mathbf{IR}}$ *over implicit* CGSs *is* **PSPACE**-*complete with respect to the size of the model and the length of the formula.*

A summary of complexity results for the alternative representation/measure of the input is presented in Figure 5.5. It turns out that, when considering the finer-grained representation that comes along with a measure based on the number of states, agents, and an encoded transition function rather than just the number of transitions, the complexity of model checking $\mathcal{L}_{ATL}$ seems distinctly harder than before for games with perfect information, and only somewhat harder for imperfect information. In particular, the problem falls into the same complexity classes for imperfect and perfect information analysis, which is rather surprising, considering the results from Section 10.1. Finally, the change of perspective does not influence the complexity of model checking of $\mathcal{L}_{ATL^*}$ as well as $\mathcal{L}_{ATL^+}$ at all.

# Part II

# Rational Agents: Models and Logics

# 6

# Rational Agents: Perfect Information

## Contents

*What can rational agents enforce and how to model rational play?* This is one of the main question addressed in this section. ATLs presented in Section 2.2 allow to express that a group of agents is able to *bring about* $\psi$. All possible behaviours are taken into account. However, such a statement is weaker than it seems. Often, we know that agents behave according to some rationality assumptions, they are not completely dumb. Therefore we do not have to check *all possible plays* – only those that are *plausible* in some reasonable sense. This has striking similarities to non-monotonic reasoning, where one considers *default rules* that describe the most plausible behaviour and allow to draw conclusions when knowledge is incomplete. In the presented approach, some strategies (or rather *strategy profiles*) can be assumed plausible, and one can reason what can be *plausibly* achieved by agents under such an assumption. Therefore, we extend **ATL** by the notion of *plausibility*, and call the resulting logic **ATLP** (*alternating time temporal logic with plausibility*). We claim that this logic is suitable to model and to reason about rational behaviour of agents.

As for **ATL** the resulting logics obtained by adding the concept of plausibility reason about the group of agents at stake. That is, given $\langle\langle A\rangle\rangle\varphi$, whether the agents in $A$ have a winning strategy for ensuring $\varphi$. However, this operator accounts only for the *theoretical existence* of such a strategy, not taking into account whether the coalition $A$ can be actually formed. Indeed, in order to join a coalition, agents usually require some kind of *incentive* (e.g. sharing common goals, getting rewards, etc.), since usually forming a coalition does not come for free (fees have to be paid, communication costs may occur, etc.). To address this important aspect we extend **ATL** by another dimension which takes the coalition formation process into account. For this purpose we combine the naive argumentative approach to coalition formation from Section 4.2 with **ATL** and call the resulting logic **CoalATL** (*coalitional alternating time temporal logic*). The new construct $\langle\!|A|\!\rangle\varphi$ denotes that *the group $A$ of agents is able to build a coalition $B$, $A \cap B \neq \emptyset$ if $A \neq \emptyset$, such that $B$ can enforce* $\varphi$. That is, it is assumed that agents in $A$ work together and try to form a coalition $B$.

The main content of this section is summarised as follows:

1. We extend **ATL** to a new logic **ATLP**$^{\textsf{base}}$ that allows to reason about what agents can achieve under an arbitrary plausibility assumption. Several expressiveness properties are shown. (These results are based on: [Bulling et al., 2009b; Jamroga and Bulling, 2007a].)

2. We present more expressive versions of **ATLP**$^{\textbf{base}}$: **ATLP**$^k$ and **ATLP**. (These results are based on: [Bulling et al., 2009b; Bulling and Jamroga, 2007b].)
3. It is shown how classical and general solution concepts (*Nash equilibrium*, *subgame perfect Nash equilibrium*, *Pareto optimality*, and others) can be characterised in the object language of **ATLP**. (These results are based on: [Bulling et al., 2009b; Jamroga and Bulling, 2007b; Bulling and Jamroga, 2007b].)
4. We turn our focus on the abilities of *sensible* coalitions and propose an naive way how this can be added to **ATL** resulting in the new logic **CoalATL**. (These results are based on: [Bulling et al., 2009a; Bulling and Dix, 2008; Bulling et al., 2008; Bulling and Dix, 2010].)

## 6.1 Reasoning About Rational Play

Agents have limited ability to predict the future. However, some lines of action seem often more sensible or realistic than others. If a rationality criterion is available, we obtain means to focus on a proper subset of possible plays. In game theoretic terms, *we solve the game*, i.e., we determine the most plausible plays, and compute their outcome. In game theory, the outcome consists of the payoffs (or utilities) assigned to players at the end of the game. In temporal logics, the outcome of a play can be seen in terms of temporal patterns that can occur — which allows for much subtler descriptions. In Section 3.2.5 we have explained how rationality can be characterised with formulae of modal logic (**ATLI** in this case). Now we show how the outcome of rational play can be described with a similar (but richer) logic, and that both aspects can be seamlessly combined.

Our logic **ATLP** (*alternating time temporal logic with plausibility*) comes in several steps, based on different underlying languages. In this section, we introduce the *base language*.

$\mathcal{L}^{base}_{ATLP}$: Sets of plausible/rational strategy profiles can be only referred to via atomic plausibility terms (constants) whose interpretation is "hardwired" in the model. A typical $\mathcal{L}^{base}_{ATLP}$ statement is $(\textbf{set-pl}\ \omega)\textbf{Pl}\,\varphi$: *Suppose that the set of rational strategy profiles is defined by* $\omega$ *– then, it is plausible to expect that* $\varphi$ *holds.* For instance, one can reason about what should happen if only Nash equilibria were played, or about the abilities of players who play only Pareto optimal profiles, had terms for Nash equilibria and Pareto optimal strategies been included in the model.

The language $\mathcal{L}^{base}_{ATLP}$ is presented in Sections 6.1.2 and the semantics in Section 6.1.4.

Then, in Section 6.2 we introduce the full logic **ATLP**. We start with an intermediate step in Section 6.2.1, namely plausibility terms written in $\mathcal{L}_{ATLI}$. This logic is named **ATLP$^{\mathbf{ATLI}}$**. It serves as a motivation to extend $\mathcal{L}_{ATLP}^{base}$ to $\mathcal{L}_{ATLP}^{1}$, and, more generally, to a hierarchy $\mathcal{L}_{ATLP} = \lim_{k\to\infty} \mathcal{L}_{ATLP}^{k}$ which denotes the language of **ATLP**.

### 6.1.1 The Concept of Rational Behaviour, Related Work

The idea has been inspired by the way in which games are analysed in game theory. Firstly, game theory identifies a number of *solution concepts* (e.g., Nash equilibrium, undominated strategies, Pareto optimality) that can be used to define rational behaviour of players. Secondly, it is usually *assumed that players play rationally* in the sense of one of the above concepts, and it is *asked about the outcome of the game under this assumption*.

In general, plausibility can be seen as a broader notion than rationality: One may obtain plausibility specifications e.g. from learning or folk knowledge. In this section, however, we mostly focus on plausibility as rationality in a *game-theoretical sense*.

There are two possible points of focus in this context. Research within game theory understandably favours work on the *characterisation* of various types of rationality (and defining most appropriate solution concepts). Applications of game theory, also understandably, tend toward *using* the solution concepts in order to predict the outcome in a given game (in other words, to "solve" the game).

The first issue has been studied in the framework of logic, for example in [Bacharach, 1987; Bonanno, 1991; Stalnaker, 1994, 1996]; more recently, game-theoretical solution concepts have been characterised in dynamic logic [Harrenstein et al., 2002, 2003], dynamic epistemic logic [Baltag, 2002; van Benthem, 2003], and extensions of **ATL** [van der Hoek et al., 2005a; Jamroga et al., 2005].

The second thread seems to have been neglected in logic-based research: The work [van Otterloo et al., 2004; van der Hoek et al., 2004; van Otterloo and Roy, 2005; van Otterloo and Jonker, 2004] are the only exceptions we know of. Moreover, each proposal from [van Otterloo et al., 2004; van der Hoek et al., 2004; van Otterloo and Roy, 2005; van Otterloo and Jonker, 2004] commits to a particular view of rationality (Nash equilibria, undominated strategies etc.). Here, we try to generalise this kind of reasoning in a way that allows to "plug in" *any* solution concept of choice (that we are able to formalise). We also try to fill in the gap between the two threads by showing how sets of rational strategy profiles can be specified in the object language, and building upon the existing work on modal logic characterisations of solution concepts [Harrenstein et al., 2002, 2003; Baltag, 2002; van Benthem, 2003; van der Hoek et al., 2005a; Jamroga et al., 2005]. We show that our logic can describe all the solution concepts that these existing approaches can describe but also additional ones, e.g. dominant strategies were it seems fundamental to allow some kind of quantification over strategy profiles.

### 6.1.2 The Language $\mathcal{L}_{ATLP}^{\mathbf{base}}$

We start with extending the language of $\mathcal{L}_{ATL}$ (cf. Definition 2.15) with operators $\mathbf{Pl}_A$, $(\mathbf{set\text{-}pl}\ \omega)$, and $(\mathbf{refn\text{-}pl}\ \omega)$. The first assumes plausible behaviour of agents in $A$; the latter are used to fix the actual meaning of plausibility by *plausibility terms* $\omega$. As yet, the terms are simple constants with no internal structure. Their meaning will be given later by a denotation function linking plausibility terms to sets of strategy profiles. Finally, the last operator is used to refine a given notion of plausibility.

**Definition 6.1 ($\mathcal{L}_{ATLP}^{\mathbf{base}}$).** *The base language $\mathcal{L}_{ATLP}^{base}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ is defined over non-empty and finite sets: $\Pi$ of propositions , $\mathbb{A}\mathrm{gt}$ of agents, and $\Omega$ of plausibility terms. $\mathcal{L}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$-formulae are defined by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle \bigcirc \varphi \mid \langle\langle A\rangle\rangle \Box\varphi \mid \langle\langle A\rangle\rangle \varphi\mathcal{U}\varphi \mid$$
$$\mathbf{Pl}_A\, \varphi \mid (\boldsymbol{set\text{-}pl}\ \omega)\varphi \mid (\boldsymbol{refn\text{-}pl}\ \omega)\varphi.$$

*Additionally, we define $\Diamond\varphi$ as $\top\mathcal{U}\varphi$, $\mathbf{Pl}$ as $\mathbf{Pl}_{\mathbb{A}\mathrm{gt}}$, and $\mathbf{Ph}$ as $\mathbf{Pl}_{\emptyset}$.*

We use $\mathsf{p}, a, \omega$ to refer to typical elements of $\Pi, \mathbb{A}\mathrm{gt}, \Omega$, respectively, and $A$ to refer to a group of agents.

$\mathbf{Pl}_A$ assumes that agents in $A$ play rationally; this means that the agents can only use strategy profiles that are *plausible* in the given model. In particular, $\mathbf{Pl}$ ($\equiv \mathbf{Pl}_{\mathbb{A}\mathrm{gt}}$) imposes rational behaviour on all agents in the system. Similarly, $\mathbf{Ph}$ disregards plausibility assumptions, and refers to all *physically* available scenarios. The model update operator $(\mathbf{set\text{-}pl}\ \omega)$ allows to define (or redefine) the set of plausible strategy profiles (referred to by $\Upsilon$ in the model) to the ones described by plausibility term $\omega$ (in this sense, it implements *revision* of plausibility). Operator $(\mathbf{refn\text{-}pl}\ \sigma)$ enables *refining* the set of plausible strategy profiles, i.e. selecting a subset of the previously plausible profiles.

With $\mathcal{L}_{ATLP}^{base}$, we can for example say that

$$\mathbf{Pl}\,\langle\langle\emptyset\rangle\rangle\Box(\mathsf{closed} \wedge \mathbf{Ph}\,\langle\langle guard\rangle\rangle \bigcirc \neg\mathsf{closed}):$$

*It is plausible to expect that the emergency door will always remain closed, but the guard retains the physical ability to open it*; or

$$(\mathbf{set\text{-}pl}\ \omega_{NE})\mathbf{Pl}\,\langle\langle 2\rangle\rangle \Diamond\mathsf{money}_2:$$

*Suppose that only playing Nash equilibria is rational; then, agent a can plausibly reach a state where it gets some money.*

We note that, in contrast to [Friedman and Halpern, 1994; Su et al., 2005; Bulling and Jamroga, 2006], the concept of plausibility presented in this article is *objective*, i.e. it does not vary from agent to agent. This is very much in the spirit of game theory, where rationality criteria are used in an analogous way. Moreover, it is *global*, because plausibility sets do not depend on the state of the system. We note, however, that the denotation of plausibility terms depends on the actual state.

### 6.1.3 Concurrent Game Structures with Plausibility

To define the semantics of $\mathcal{L}^{base}_{ATLP}$ (and also $\mathcal{L}_{ATLP}$ introduced in the following section), we extend CGSs to *concurrent game structures with plausibility*. Apart from an actual set of plausible strategies $\Upsilon$, a *concurrent game structure with plausibility* (CGSP) must specify the denotation of plausibility terms $\omega \in \Omega$. It is defined via a *plausibility mapping*

$$[\![\cdot]\!] : Q \to (\Omega \to \mathcal{P}(\Sigma)).$$

Instead of $[\![q]\!](\omega)$ we will often write $[\![\omega]\!]^q$ to turn the focus to the plausibility terms. Each term is mapped to a *set* of strategy profiles. We note also, that the denotation of a term depends on the state. In a way, the current state of the system defines the "initial position in the game", and this heavily influences the set of rational strategy profiles for most rationality criteria. For example, a strategy profile can be a Nash equilibrium (NE) in $q_0$, and yet it may not be a NE in some of its successors.

We will propose a more concrete (and more practical) implementation of plausibility terms in Section 6.2.1 and 6.2.2.

**Definition 6.2** (CGSP)**.** *A* concurrent game structure with plausibility (CGSP) *is given by a tuple*

$$\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \Upsilon, \Omega, [\![\cdot]\!] \rangle$$

*where* $\langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o \rangle$ *is a* CGS *(cf. Definition 2.16),* $\Upsilon \subseteq \Sigma$ *is a set of plausible strategy profiles (called* plausibility set*);* $\Omega$ *is a set of of plausibility terms, and* $[\![\cdot]\!]$ *is a plausibility mapping over* $Q$ *and* $\Omega$*.*

*By* $CGSP(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ *we denote the set of all* CGSPs *over* $\mathbb{A}\mathrm{gt}$*,* $\Pi$ *and* $\Omega$*. Furthermore, for a given* CGSP $\mathfrak{M}$ *we use* $X_{\mathfrak{M}}$ *to refer to element* $X$ *of* $\mathfrak{M}$*, e.g.,* $Q_{\mathfrak{M}}$ *to refer to the set* $Q$ *of states of* $\mathfrak{M}$*.*

**Definition 6.3 (Compatible model).** *Given a formula* $\varphi \in \mathcal{L}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ *a* CGSP $\mathfrak{M}$ *is called* compatible *with* $\varphi$ *if, and only if,* $\mathfrak{M} \in CGSP(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$*. That is, the model interprets all symbols occurring in* $\varphi$*. A model* $\mathfrak{M}$ *is called compatible with a set* $\mathcal{L}$ *of* $\mathcal{L}_{ATLP}$*-formulae if, and only if,* $\mathfrak{M}$ *is compatible with each formula in* $\mathcal{L}$*.*

*We will assume by default that, given a formula or a set of formulae, the model we consider is compatible with it.*

The formula $\mathbf{Pl}\langle\langle A\rangle\rangle\gamma$ implies that $A$ can only play plausible strategies. Thus, $A$'s part of the strategy profiles in $\Upsilon$ is of particular interest which motivates the following definition.

**Definition 6.4 ($P_B|_A$).** *For a given set $P_B \subseteq \Sigma_B$ of collective strategies of agents $B$, $P_B|_A$ with $A \neq \emptyset$ denotes the set of $A$'s substrategies in $P_B$, i.e.:*

$$P_B|_A = \{s_A \in \Sigma_A \mid \exists s'_B \in P_B \quad (s'_B|_A = s_A)\}.$$

*For $A = \emptyset$ we define $P_B|_\emptyset$ to consist of the empty strategy; i.e.,*

$$P_B|_\emptyset = \{s_\emptyset\}.$$

In particular, if $\emptyset \neq A \not\subseteq B$ we have that $P_B|_A = \emptyset$. (We would like to note that this is different from $\{s_\emptyset\}$.)

Often, we impose restrictions only on a subset $B \subseteq \mathbb{A}\mathrm{gt}$ of agents, without assuming rational play of all agents. This can be desirable due to several reasons. It might, for example, be the case that only information about the proponents' play is available; hence, assuming plausible behaviour of the opponents is neither sensible nor justified. Or, even simpler, a group of (simple minded) agents might be known to not behave rationally.

Consider formula $\mathbf{Pl}_B\langle\langle A\rangle\rangle\gamma$: The team $A$ looks for a strategy that brings about $\gamma$, but the members of the team who are also in $B$ can only choose plausible strategies. The same applies to $A$'s opponents that are contained in $B$. Strategies which comply with $B$'s part of some plausible strategy profile are called $B$-plausible.

**Definition 6.5 ($B$-plausibility of strategies).** *Let $A, B \subseteq \mathbb{A}\mathrm{gt}$ and $s_A \in \Sigma_A$. We say that $s_A$ is $B$-*plausible in $\mathfrak{M}$ *if, and only if, $B$'s substrategy in $s_A$ is part of some plausible strategy profile in $\mathfrak{M}$, i.e., if $s_A|_{A\cap B} \in \Upsilon_{\mathfrak{M}}|_{A\cap B}$.*

*By $\Upsilon_{\mathfrak{M}}(B)$ we denote the set of all $B$-plausible strategy profiles in $\mathfrak{M}$. That is, $\Upsilon_{\mathfrak{M}}(B) = \{s \in \Sigma \mid s|_B \in \Upsilon_{\mathfrak{M}}|_B\}$. Note that $s_A$ is $B$-plausible iff $s_A \in \Upsilon_{\mathfrak{M}}(B)|_A$.*

We observe that $s_A$ is trivially $B$-plausible whenever $A$ and $B$ are disjoint.

*Remark 6.6.* We note, that the set of $\emptyset$-plausible strategy profiles is given by the set of all strategy profiles (cf. Definition 6.4):

$$\Upsilon(\emptyset) = \{s \in \Sigma \mid s|_\emptyset \in \Upsilon|_\emptyset\} = \{s \in \Sigma \mid \emptyset \in \{\emptyset\}\} = \Sigma$$

and for $\Upsilon = \emptyset$ we have that $\Upsilon(A) = \emptyset$ for $A \neq \emptyset$.

As mentioned above, if some opponents belong to the set of agents who are assumed to play plausibly then they must also comply with the actual plausibility specifications when choosing their actions; this is taken into account by the following notion of plausible outcome.

**Definition 6.7 ($B$-plausible outcome).** *The* $B$-plausible outcome*,*

$$out_{\mathfrak{M}}(q, s_A, B),$$

*with respect to strategy $s_A$ and state $q$ is defined as the set of paths which can occur when only $B$-plausible strategy profiles can be played and agents in $A$ follow $s_A$:*

$$out_{\mathfrak{M}}(q, s_A, B) = \{\lambda \in \Lambda_{\mathfrak{M}}(q) \mid \text{ there exists a } B\text{-plausible } t \in \Sigma \text{ such that } t|_A = s_A \text{ and } out_{\mathfrak{M}}(q, t) = \{\lambda\}\}.$$

*As before we will leave out the subscript "$\mathfrak{M}$" if clear from context.*

We note that the outcome $out_{\mathfrak{M}}(q, s_A, B)$ is empty whenever $(A \cap B)$'s part of $s_A$ is not part of any plausible strategy profile in $\Upsilon_{\mathfrak{M}}$. For example, assume that all agents in $B$ play only parts of Nash equilibria. Then for a given $s_A$ there are two possibilities for the $B$-consistent outcome. Either it is empty because $(A \cap B)$'s part of $s_A$ does not belong to any Nash equilibrium, or it consists of all paths which can occur when (1) $A$ sticks to $s_A$, (2) $B$ (including $A \cap B$) plays according to some Nash equilibrium, and (3) the other agents behave arbitrarily.

### 6.1.4 Semantics: The Logic ATLP$^{\text{base}}$

The truth of $\mathcal{L}_{ATLP}^{base}$-formulae is given with respect to a CGSP, a state $q$, and a set $B$ of agents. The intuitive reading of $\mathfrak{M}, q \models_B \varphi$ is: $\varphi$ is true in model $\mathfrak{M}$ and state $q$ if it is assumed that players in $B$ play rationally (i.e., by using only plausible combinations of strategies). No constraints are imposed on the behaviour of agents outside $B$, but the plausibility operator $\mathbf{Pl}_A$ can be used to change the set of agents (viz. $A$) whose play is restricted. The update/refinement modalities $(\mathbf{set\text{-}pl}\ \omega)$/$(\mathbf{refn\text{-}pl}\ \omega)$ are used to change the plausibility set $\Upsilon_{\mathfrak{M}}$ in the model.

**Definition 6.8 (Semantics of $\mathcal{L}_{\boldsymbol{ATLP}}^{\mathbf{base}}$).** *Let $\mathfrak{M} \in CGSP(\mathbb{A}\text{gt}, \Pi, \Omega)$ and $A, B \subseteq \mathbb{A}\text{gt}$. The semantics of $\mathcal{L}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \Omega)$-formulae is given as follows:*

$\mathfrak{M}, q \models_B \mathsf{p}$ *iff* $\mathsf{p} \in \pi(q)$ *and* $\mathsf{p} \in \Pi$*;*
$\mathfrak{M}, q \models_B \neg\varphi$ *iff* $\mathfrak{M}, q \not\models_B \varphi$*;*
$\mathfrak{M}, q \models_B \varphi \wedge \psi$ *iff* $\mathfrak{M}, q \models_B \varphi$ *and* $\mathfrak{M}, q \models_B \psi$*;*

$\mathfrak{M}, q \models_B \langle\langle A\rangle\rangle \bigcirc \varphi$ *iff there is a B-plausible* $s_A$ *such that* $\mathfrak{M}, \lambda[1] \models_B \varphi$ *for all* $\lambda \in out_{\mathfrak{M}}(q, s_A, B)$;

$\mathfrak{M}, q \models_B \langle\langle A\rangle\rangle \Box\varphi$ *iff there is a B-plausible* $s_A$ *such that* $\mathfrak{M}, \lambda[i] \models_B \varphi$ *for all* $\lambda \in out_{\mathfrak{M}}(q, s_A, B)$ *and all* $i \in \mathbb{N}_0$;

$\mathfrak{M}, q \models_B \langle\langle A\rangle\rangle \varphi\mathcal{U}\psi$ *iff there is a B-plausible* $s_A$ *such that, for all* $\lambda \in out_{\mathfrak{M}}(q, s_A, B)$, *there is* $i \in \mathbb{N}_0$ *with* $\mathfrak{M}, \lambda[i] \models_B \psi$, *and* $\mathfrak{M}, \lambda[j] \models_B \varphi$ *for all* $0 \leq j < i$;

$\mathfrak{M}, q \models_B \mathbf{Pl}_A\, \varphi$ *iff* $\mathfrak{M}, q \models_A \varphi$;

$\mathfrak{M}, q \models_B (\textbf{\textit{set-pl}}\, \omega)\varphi$ *iff* $\mathfrak{M}', q \models_B \varphi$ *where the new model* $\mathfrak{M}'$ *is equal to* $\mathfrak{M}$ *but the new set* $\Upsilon_{\mathfrak{M}'}$ *of plausible strategy profiles of* $\mathfrak{M}'$ *is set to* $[\![\omega]\!]^q_{\mathfrak{M}}$;

$\mathfrak{M}, q \models_B (\textbf{\textit{refn-pl}}\, \omega)\varphi$ *iff* $\mathfrak{M}', q \models_B \varphi$ *where* $\mathfrak{M}'$ *is equal to* $\mathfrak{M}$ *but* $\Upsilon_{\mathfrak{M}'}$ *set to* $\Upsilon_{\mathfrak{M}} \cap [\![\omega]\!]^q_{\mathfrak{M}}$.

*The "absolute" satisfaction relation* $\models$ *is given by* $\models_\emptyset$.

**Definition 6.9 (Validity).** *Let* $\varphi \in \mathcal{L}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ *and* $M \subseteq CGSP(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ *be a set of models. Formula* $\varphi$ *is* valid *with respect to* $M$ *if, and only if,* $\mathfrak{M}, q \models \varphi$ *for each* $\mathfrak{M} \in M$ *and each state* $q \in Q_{\mathfrak{M}}$.

An ordinary concurrent game structure (without plausibility) can be interpreted as a CGSP with all strategy profiles assumed plausible, i.e., with $\Upsilon = \Sigma$, and empty set of plausibility terms $\Omega$.

Let us clarify the semantics behind $\mathbf{Pl}_B \langle\langle A\rangle\rangle\gamma$ once more. The proponents (viz. $A$) look for a strategy that enforces $\gamma$; some of them (viz. $A \cap B$) are assumed to play a part of a plausible strategy profile while the others (viz. $A \setminus B$) can choose an arbitrary collective strategy. Analogously, some opponents (viz. $B \setminus A$) are supposed to play plausibly (their choice complies to set $\Upsilon_{\mathfrak{M}}$ together with the strategy already chosen by $A \cap B$), while the rest (viz. $\mathbb{A}\mathrm{gt} \setminus (A \cup B)$) has unrestricted choice. In particular, when $B = A$, only the choices of the proponents are restricted; for $B = \mathbb{A}\mathrm{gt} \setminus A$ plausibility restrictions apply to the opponents only.

*Remark 6.10.* We observe that our framework is semantically similar to the approach of *social laws* [Shoham and Tennenholz, 1992; Moses and Tennenholz, 1995; van der Hoek et al., 2005b]. However, we refer to *strategy profiles* as rational or not, while social laws define constraints on agents' *individual actions.* Also, our motivation is different: In our framework, agents are expected to behave in a specified way because it is rational in some sense; social laws prescribe behaviour sanctioned by social norms and legal regulations.

*Example 6.11 (Asymmetric matching pennies ctd.).* We continue our Example 3.10. Suppose that it is plausible to expect that both agents are rational in the sense that they only play undominated strategies.[1] Then, $\Upsilon =$

[1] We recall from Section 3.1 that a strategy $s_a \in \Sigma_a$ is called *undominated* if, and only if, there is no strategy $s'_a \in \Sigma_a$ such that the achieved utility of $s'_a$ is at least

$\{(s_h, s_h), (s_h, s_t)\}$. Under this assumption, agent 2 is free to grant itself with the prise or to refuse it: $\mathbf{Pl}\,(\langle\!\langle 2\rangle\!\rangle\Diamond\ \mathsf{money}_2 \wedge \langle\!\langle 2\rangle\!\rangle\Box\neg\mathsf{money}_2)$. Still, it cannot choose to win without making the other player win too: $\mathbf{Pl}\,\neg\langle\!\langle 2\rangle\!\rangle\Diamond(\mathsf{money}_2 \wedge \neg\mathsf{money}_1)$. Likewise, if rationality is defined via iterated undominated strategies, then we have $\Upsilon = \{(s_h, s_h)\}$, and therefore the outcome of the game is completely determined: $\mathbf{Pl}\,\langle\!\langle\emptyset\rangle\!\rangle\Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1 \wedge \mathsf{money}_2)$.

In order to include *both* notions of rationality in the model, we can encode them as denotations of two different plausibility terms – say, $\omega_{undom}$ and $\omega_{iter}$, with $[\![\omega_{undom}]\!]^{q_0} = \{(s_h, s_h), (s_h, s_t)\}$, and $[\![\omega_{iter}]\!]^{q_0} = \{(s_h, s_h)\}$. Let $\mathfrak{M}'_1$ be model $\mathfrak{M}_1$ with plausibility terms and their denotation defined as above. Then, we have that $\mathfrak{M}'_1, q_0 \models (\mathbf{set\text{-}pl}\ \omega_{undom})\mathbf{Pl}\,(\langle\!\langle 2\rangle\!\rangle\Diamond\ \mathsf{money}_2 \wedge \langle\!\langle 2\rangle\!\rangle\Box\neg\mathsf{money}_2) \wedge (\mathbf{set\text{-}pl}\ \omega_{iter})\mathbf{Pl}\,\langle\!\langle\emptyset\rangle\!\rangle\Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1 \wedge \mathsf{money}_2)$.

Out of many solution concepts, Nash equilibria are the most widely accepted, especially for non-cooperative games. We briefly extend our working example with game analysis based on Nash equilibria. In this case, it is not possible to define rationality with independent constraints on agents' individual strategies (like in normative systems). These are full strategy profiles being rational or not, since rationality of a strategy depends on the actual response of the other players.

*Example 6.12 (Asymmetric matching pennies ctd.).* Suppose that rationality is defined through Nash equilibria. Then, $\Upsilon = \{(s_h, s_h), (s_t, s_t)\}$. Under this assumption, agent 2 is sure to get the prise: $\mathbf{Pl}\,\langle\!\langle\emptyset\rangle\!\rangle\Box(\neg\mathsf{start} \rightarrow \mathsf{money}_2)$.

Moreover, by choosing the right strategy, 2 can control the outcome of the other agent: $\mathbf{Pl}\,(\langle\!\langle 2\rangle\!\rangle\Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1) \wedge \langle\!\langle 2\rangle\!\rangle\Box\neg\mathsf{money}_1)$. Note that agent 1 can control its own outcome too, if we assume that the players are obliged to play rationally: $\mathbf{Pl}\,(\langle\!\langle 1\rangle\!\rangle\Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1) \wedge \langle\!\langle 1\rangle\!\rangle\Box\neg\mathsf{money}_1)$. This may seem strange, but a Nash equilibrium assumes implicitly that the agents coordinate their actions somehow. Then, assuming a particular choice of one agent in advance *constrains* the other agent responses considerably, which puts the first agent at advantage.

*Example 6.13 (Bargaining ctd.).* We continue Example 3.12. Let $\omega_{NE}$ denote the set of Nash equilibria (every payoff can be reached by a Nash equilibrium), and $\omega_{SPN}$ the set of subgame perfect Nash equilibria in the game. Then, the following holds for every $x \in [0, 1]$:

$$\mathfrak{M}'_2, q_0 \models$$
$$(\mathbf{set\text{-}pl}\ \omega_{NE})\langle\!\langle 1,2\rangle\!\rangle\Diamond(\mathsf{p}_1^{\mathsf{x}} \wedge \mathsf{p}_2^{1-\mathsf{x}}) \wedge (\mathbf{set\text{-}pl}\ \omega_{SPN})\langle\!\langle\emptyset\rangle\!\rangle\Diamond(\mathsf{p}_1^{\frac{1-\delta_2}{1-\delta_1\delta_2}} \wedge \mathsf{p}_2^{\frac{\delta_2(1-\delta_1)}{1-\delta_1\delta_2}}).$$

as good as for $s_a$ for all counterstrategies $s_{-a} \in \Sigma_{\mathbb{A}\mathrm{gt}\setminus\{a\}}$ and strictly better for at least one counterstrategy $s_{-a} \in \Sigma_{\mathbb{A}\mathrm{gt}\setminus\{a\}}$.

where $\mathfrak{M}'_2$ is given by $\mathfrak{M}_2$ extended by plausibility terms and their denotation as introduced above.

Finally, we observe that the "plausibility refinement" operator (**refn-pl** $\cdot$) can be used to combine several solution concepts, e.g., (**set-pl** $\omega_{NE}$) (**refn-pl** $\omega_{PO}$) restricts plausible play to *Pareto optimal Nash equilibria*. We can also use (**refn-pl** $\cdot$) to *compare* different notions of rationality. For example,

$$(\textbf{set-pl}\ \omega_{NE})(\textbf{refn-pl}\ \omega_{PO})\langle\langle \mathbb{A}\mathrm{gt}\rangle\rangle \bigcirc \top$$

can be used to check if Pareto optimal strategies that are also Nash equilibria exist in the model at all.

The base language $\mathcal{L}^{base}_{ATLP}$ allows to restrict the analysis to a subset of available strategy profiles. One drawback of $\mathcal{L}^{base}_{ATLP}$ is that we cannot specify sets of plausible/rational strategy profiles *in the object language*, simply because our terms do not have any internal structure — they are just constants. Ideally, one would like to have a flexible language of terms that allows to specify *any sensible rationality assumption*, and then impose it on the system.

Our first step is to employ $\mathcal{L}_{ATLI}$-formulae and make use of the results in Section 3.2.5. The second step is to define a proper extension of $\mathcal{L}^{base}_{ATLP}$ where these concepts can be expressed, thus enabling both specification of plausibility and reasoning about plausible behaviour to be conducted in **ATLP**. The idea is to use $\mathcal{L}_{ATLP}$ formulae $\theta$ themselves to specify sets of plausible strategy profiles, with the intended meaning that $\Upsilon$ collects exactly the profiles for which $\theta$ holds. Then, we can embed such an $\mathcal{L}_{ATLP}$-based plausibility specification in another formula of $\mathcal{L}_{ATLP}$.

## 6.2 Alternating-Time Temporal Logic with Plausibility: ATLP

In this section we extend the language $\mathcal{L}^{base}_{ATLP}$ and present the following languages:

$\mathcal{L}^{0}_{ATLP}$: A slight extension of $\mathcal{L}^{base}_{ATLP}$. We allow some combinations of the constants of $\mathcal{L}^{base}_{ATLP}$ to form more complex terms.

$\mathcal{L}^{ATLI}_{ATLP}$: An intermediate language, where rational strategy profiles are characterised by $\mathcal{L}_{ATLI}$-formulae.

$\mathcal{L}^{k}_{ATLP}$: Here we have nestings of plausibility updates up to level $k$. It turns out that the logic over $\mathcal{L}^{ATLI}_{ATLP}$ is already embedded in the logic over $\mathcal{L}^{1}_{ATLP}$.

$\mathcal{L}_{ATLP}$: Unbounded nestings of formulae are allowed.

The full logic **ATLP** (*alternating time temporal logic with plausibility*) is based on the last, most expressive language.

### 6.2.1 Plausibility Terms based on $\mathcal{L}_{ATLI}$

In Section 3.2.5 we have recalled the logic **ATLI**, a logic suitable to characterise game-theoretic solution concepts. Here, we take such formulae and use them as the strategic terms used in $\mathcal{L}^{base}_{ATLP}$.

**Definition 6.14 (The language $\mathcal{L}^{ATLI}_{ATLP}$, $\Omega^*$).** *Let*

$$\Omega^* = \{(\sigma.\theta) \mid \theta \in \mathcal{L}_{ATLI}(\mathbb{A}\mathrm{gt}, \Pi, \{\sigma[1], \dots, \sigma[k]\})\}.$$

*That is, $\Omega^*$ collects terms of the form $(\sigma.\theta)$, where $\theta$ is an $\mathcal{L}_{ATLI}$-formula including only a single strategic term $\sigma[a]$ for each agent $a$. The language $\mathcal{L}^{ATLI}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi)$ is defined as $\mathcal{L}^{base}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega^*)$.*

The idea underlying terms is as follows. We have an $\mathcal{L}^{ATLI}_{ATLP}$-formula $\theta$, parameterised with a variable $\sigma$ that ranges over the set of strategy profiles $\Sigma$. Now, we want $(\sigma.\theta)$ to denote exactly the set of profiles from $\Sigma$, for which formula $\theta$ holds. However – as $\sigma$ denotes a strategy profile, and **ATLI** allows only to refer to strategies of individual agents – we need a way of addressing substrategies of $\sigma$ in $\theta$. This can be done by using $\mathcal{L}_{ATLI}$-terms $\sigma[i]$, which are interpreted as $i$'s substrategy in $\sigma$.

For example, we may assume that a rational agent does not grant other agents with too much control over its life:

$$(\sigma \,.\, \bigwedge_{a \in \mathbb{A}\mathrm{gt}} ((\mathbf{str}_a \sigma[a]) \neg \langle\!\langle \mathbb{A}\mathrm{gt} \setminus \{a\} \rangle\!\rangle \Diamond \mathsf{dead_a})).$$

Games defined by CGSs are, in general, not determined, so the above specification does not guarantee that each rational agent can efficiently protect its life. It only requires that it should behave cautiously so that its opponents do not have complete power to kill it.

**Definition 6.15 (Denotation of $\mathcal{L}_{ATLI}$-based plausibility terms).** *Let* $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o \rangle$ *be a* CGS *and* $\Omega^*$ *be as in Definition 6.14. For each* $s \in \Sigma$ *we define* $\mathfrak{M}^s$ *to be the following* CGS *with intentions:*

$$\mathfrak{M}^s = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \mathcal{I}^0, \mathfrak{Str}, [\cdot] \rangle$$

*with* $\mathfrak{Str}_a = \{\sigma[a]\}$, $[\sigma[a]] = s|_a$, *and* $\mathfrak{Str} = \bigcup_{a \in \mathbb{A}\mathrm{gt}} \mathfrak{Str}_a$. *We recall from Section 3.2.5 that* $\mathcal{I}^0$ *represents the full intention relation.*

*The plausibility mapping for terms from* $\Omega^*$ *is defined as:*

$$[\![\sigma.\theta]\!]^q = \{s \in \Sigma \mid \mathfrak{M}^s, q \models_{\mathsf{ATLI}} \theta\}.$$

It is now possible to plug in arbitrary $\mathcal{L}_{ATLI}$-specifications of rationality, and to reason about their consequences.

*Example 6.16 (Asymmetric matching pennies ctd.).* It seems that explicit quantification over the opponents' responses (not available in **ATLI**) is essential to express undominatedness of strategies (cf. [van der Hoek et al., 2005a] and Section 6.4.2) if one does not want to enumerate all possible strategies explicitly. Still, we can at least assume that a rational player should avoid playing strategies that *guarantee* failure if a potentially successful strategy is available. Under this assumption, player 1 should never play tail, and as a consequence player 2 controls the outcome of the game:

$$\mathfrak{M}_1'', q_0 \models (\textbf{set-pl}\ \sigma. \textstyle\bigwedge_{a\in\mathbb{A}\mathrm{gt}}(\langle\!\langle\mathbb{A}\mathrm{gt}\rangle\!\rangle\Diamond\mathsf{money_a} \rightarrow (\mathbf{str}_a\sigma[a])\langle\!\langle\mathbb{A}\mathrm{gt}\rangle\!\rangle\Diamond\mathsf{money_a}))$$
$$\mathbf{Pl}\,\big(\langle\!\langle 2\rangle\!\rangle\Diamond(\mathsf{money_1}\wedge\mathsf{money_2})\wedge\langle\!\langle 2\rangle\!\rangle\Box\neg(\mathsf{money_1}\wedge\mathsf{money_2})\big),$$

where $\mathfrak{M}_1''$ is the CGS $\mathfrak{M}_1$ extended with propositions $\mathsf{p_i^1} \equiv \mathsf{money_i}$, $\mathcal{L}_{ATLI}$-based plausibility terms, and their denotation according to Definition 6.15.

Moreover, if only Pareto optimal strategy profiles can be played, then both players are bound to keep winning money:

$$\mathfrak{M}_1'', q_0 \models (\textbf{set-pl}\ \sigma.PO^{\Diamond}(\sigma))\ \mathbf{Pl}\,\langle\!\langle\emptyset\rangle\!\rangle\Box(\neg\mathsf{start}\rightarrow\mathsf{money_1}\wedge\mathsf{money_2}).$$

Finally, restricting plausible strategy profiles to Nash equilibria guarantees that player 2 should plausibly get money, but the outcome of player 1 is not determined:

$$\mathfrak{M}_1'', q_0 \models (\textbf{set-pl}\ \sigma.NE^{\Diamond}(\sigma))\ \mathbf{Pl}\,\big(\langle\!\langle\emptyset\rangle\!\rangle\Box(\neg\mathsf{start}\rightarrow\mathsf{money_2})$$
$$\wedge\neg\langle\!\langle\emptyset\rangle\!\rangle\Diamond\mathsf{money_1}\wedge\neg\langle\!\langle\emptyset\rangle\!\rangle\Box\neg\mathsf{money_1}\big).$$

*NE* and *PO* refer to the formulae defined in Section 3.3.1.

*Example 6.17 (Bargaining ctd.).* For the bargaining agents and $\kappa = (1-\delta_2)\frac{1-(\delta_1\delta_2)^{\frac{T}{2}}}{1-\delta_1\delta_2} + (\delta_1\delta_2)^{\frac{T}{2}}$, we have accordingly:

1. $\mathfrak{M}_2', q_0 \models (\textbf{set-pl}\ \sigma.NE^{\Diamond}(\sigma))\mathbf{Pl}\,\langle\!\langle\emptyset\rangle\!\rangle\bigcirc(\mathsf{p_1^x}\wedge\mathsf{p_2^{1-x}})$ for every $x$;
2. $\mathfrak{M}_2', q_0 \models (\textbf{set-pl}\ \sigma.SPN^{\Diamond}(\sigma))\mathbf{Pl}\,\langle\!\langle\emptyset\rangle\!\rangle\bigcirc(\mathsf{p_1^\kappa}\wedge\mathsf{p_2^{1-\kappa}})$;
3. $\mathfrak{M}_2', q_0 \models (\textbf{set-pl}\ \sigma.SPN^{\Diamond}(\sigma))\mathbf{Pl}\,\langle\!\langle\emptyset\rangle\!\rangle\Box(\neg\mathsf{p_1^{x_1}}\wedge\neg\mathsf{p_2^{x_2}})$ for every $x_1\neq\kappa$ and $x_2\neq 1-\kappa$

where $\mathfrak{M}_2'$ is the CGSP obtained from CGS $\mathfrak{M}_2$ by adding $\mathcal{L}_{ATLI}$-based plausibility terms and their denotation. Where *NE*, *PO*, and *SPN* refer to the formulae defined in Section 3.3.1.

Thus, we can encode a game as a CGS $\mathfrak{M}$, specify rationality assumptions with an $\mathcal{L}_{ATLI}$-formula $\theta$, and ask if a desired property $\varphi$ of the system holds under these assumptions by model checking $(\textbf{set-pl}\ \sigma.\theta)\varphi$ in $\mathfrak{M}$. Note that the denotation of plausibility terms in $\Omega^*$ is fixed. We report our results on the complexity of solving such games in Section 10.1.

### 6.2.2 The Languages $\mathcal{L}^k_{ATLP}$ and $\mathcal{L}_{ATLP}$

As we have already explained, our main idea is to use **ATLP** for both specification of rationality assumptions and description of the outcome of rational play. Thus, we need a possibility to embed an $\mathcal{L}_{ATLP}$-formula $\varphi$ (that defines the rationality condition) in a "higher-level" formula of $\mathcal{L}_{ATLP}$, as a part of plausibility term (**set-pl** $\sigma.\varphi$). The reading of (**set-pl** $\sigma.\varphi)\psi$ is, again: "Let the plausibility set consist of profiles $\sigma$ that satisfy $\varphi$; then, $\psi$ holds". Apart from the possibility of nesting formulae via plausibility updates, we also propose to add quantifier-like structures to the language of terms. Consider, for example, the term $\sigma_1.(\exists\sigma_2)\varphi$. We would like to *collect* all strategies $s_1$ such that there is a strategy $s_2$ for which $\varphi$ holds (we use $\sigma_i$ to refer to $s_i$). Thus, $\sigma_1.(\exists\sigma_2)\varphi$ is supposed to act in a similar way as the first order logic-based set specification $\{x \mid \exists y : \varphi(x, y)\}$. It is easy to see that e.g. the set of all undominated strategies can now be specified in a straightforward way.

As before, the full logic **ATLP** is given over a set $\mathbb{A}\mathrm{gt}$ of agents, a set $\Pi$ of propositions, and a set $\Omega$ of *primitive plausibility terms* (cf. Section 6.1.4). In addition to these sets, we also include a set $\mathcal{V}ar$ of *strategic variables*. Variables in $\mathcal{V}ar$ range over strategy profiles; we need them to characterise specific rationality criteria, in a way similar to first order logic specifications.

The definition of $\mathcal{L}_{ATLP}$ is given recursively. In each step the structure of plausibility terms becomes more sophisticated. At first, we only consider terms out of $\Omega$; their interpretation is given in the model. On the next level, we also allow plausibility terms to be quantified $\mathcal{L}_{ATLP}$-formulae which contain strategic variables *and* elements from $\Omega$. Plausibility terms of subsequent levels can again be based on terms from the previous levels, and so forth. As a consequence, the *core 0-level language* of $\mathcal{L}_{ATLP}$ is almost the same as the base language $\mathcal{L}^{base}_{ATLP}$ defined in Section 6.1.2: It only extends it with simple *combinations* of terms.

In general, all the levels of the language can be seen as containing ordinary formulae of $\mathcal{L}^{base}_{ATLP}$, the only thing that changes as we move to higher levels is the complexity of plausibility terms. We begin with defining simple combinations of plausibility terms, and then present the hierarchy of languages $\mathcal{L}^k_{ATLP}$, with the underlying idea that $\mathcal{L}^k_{ATLP}$ allows for at most $k$ ($k \in \mathbb{N}_0$) nested plausibility update operators. The full language $\mathcal{L}_{ATLP}$ allows for any arbitrary finite number of nestings. Firstly, we define how strategy profiles can be combined and mixed.

**Definition 6.18 (Strategic combination).** *Let $X$ be a non-empty set of symbols. We say that $y$ is a* strategic combination *of $X$ if it is generated by the following grammar:*

$$y ::= x \mid \langle y, \dots, y \rangle \mid y[A]$$

*where* $x \in X$*,* $\langle y, \dots, y \rangle$ *is a vector of length* $|\mathbb{A}\text{gt}|$*, and* $A \subseteq \mathbb{A}\text{gt}$*. The set of* strategic combinations *over* $X$ *is defined by* $\mathcal{T}(X)$*. It is easy to see that operator* $\mathcal{T}$ *is idempotent (i.e.* $\mathcal{T}(X) = \mathcal{T}(\mathcal{T}(X))$*).*

The intuition is that elements of $x \in X$ are symbols in the object language that refer to *sets* of strategy *profiles* (as the basic plausibility terms do), and the elements of $\mathcal{T}(X)$ allow to combine these sets to new sets.[2] Let $x$ refer to a set of strategy profiles $\chi \subseteq \Sigma$. Then, $x[A]$ refers to all the profiles in $\Sigma$ in which $A$'s substrategy is part of some profile from $\chi$. Similarly, if $x_1, \dots, x_k$ denote sets of strategy profiles $\chi_1, \dots, \chi_k$, then $\langle x_1, \dots, x_k \rangle$ refers to all the profiles that agree on agent $i$'s strategy with $i$'s part on at least one profile from $\chi_i$ for each $i = 1, \dots, k$.

**Definition 6.19 (**$\mathcal{L}^k_{ATLP}$**).** *Let* $\mathbb{A}\text{gt}$ *be a set of agents,* $\Pi$ *a set of propositions,* $\Omega$ *be a set of primitive plausibility terms, and* $\mathcal{V}ar$ *a set of* strategic variables *(with typical elements* $\sigma, \sigma_1, \sigma_2, \dots$*). The languages* $\mathcal{L}^k_{ATLP}(\mathbb{A}\text{gt}, \Pi, \mathcal{V}ar, \Omega)$*, for* $k = 0, 1, 2, \dots,$ *are recursively defined as follows:*

- $\mathcal{L}^0_{ATLP}(\mathbb{A}\text{gt}, \Pi, \mathcal{V}ar, \Omega) = \mathcal{L}^{base}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \Omega_0)$, *where* $\Omega_0 = \mathcal{T}(\Omega)$*;*
- $\mathcal{L}^k_{ATLP}(\mathbb{A}\text{gt}, \Pi, \mathcal{V}ar, \Omega) = \mathcal{L}^{base}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \Omega_k)$, *where:*
  $\Omega_k = \mathcal{T}(\Omega_{k-1} \cup \Omega^k_{new})$,
  $\Omega^k_{new} = \{\sigma_1.(Q_2\sigma_2)\dots(Q_n\sigma_n)\varphi \mid n \in \mathbb{N}, \forall i \ (1 \le i \le n \Rightarrow$
  $\sigma_i \in \mathcal{V}ar, \ Q_i \in \{\forall, \exists\}, \varphi \in \mathcal{L}^{base}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \mathcal{T}(\Omega_{k-1} \cup \{\sigma_1, \dots, \sigma_n\})))\ \}$.

Thus, plausibility terms on level $k$ (i.e., $\Omega_k$) augment terms from the previous level ($\Omega_{k-1}$) with new terms $\Omega^k_{\text{new}}$ that combine *quantification over strategic variables* $\sigma_1, \dots \sigma_n$ *with formulae possibly containing these strategic variables*. Such terms are used to *collect* (or describe) specific strategy profiles (referred to by variable $\sigma_1$ which plays a distinctive role in comparison with the other variables). In the following we will often use $\sigma$ to refer to $\sigma_1$.

**Definition 6.20 (**$\mathcal{L}_{ATLP}$**).** *The set of* $\mathcal{L}_{ATLP}$*-formulae with arbitrary finite nesting of plausibility terms is defined by*

$$\begin{aligned}\mathcal{L}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \mathcal{V}ar, \Omega) &:= \mathcal{L}^{\infty}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \mathcal{V}ar, \Omega)\\ &:= \lim_{k \to \infty} \mathcal{L}^k_{ATLP}(\mathbb{A}\text{gt}, \Pi, \mathcal{V}ar, \Omega).\end{aligned}$$

**Definition 6.21 (**$k$**-formula,** $k$**-term).** *Formula* $\varphi \in \mathcal{L}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \mathcal{V}ar, \Omega)$ *is called an* $\mathcal{L}^k_{ATLP}$*-formula (or simply* $k$-formula*) if, and only if,* $\varphi \in \mathcal{L}^k_{ATLP}(\mathbb{A}\text{gt}, \Pi, \mathcal{V}ar, \Omega)$*. Analogously, a plausibility term occurring in a* $k$*-formula is called a* $k$-(plausibility) term.

[2] This correspondence will be given formally in Definition 6.24 (Section 6.2.3).

*Remark 6.22.* We use the acronym ATLP to refer to both the full logic **ATLP**, the base logic **ATLP^base** and so on.

*Example 6.23 (Illustration of plausibility terms in $\mathcal{L}^{k}_{ATLP}$).* Below we present some simple formulae illustrating the different levels of our logic.

$\mathcal{L}^{base}_{ATLP}$: $(\mathbf{set\text{-}pl}\ \omega_{\mathrm{NE}})\mathbf{Pl}\,\langle\!\langle A\rangle\!\rangle\gamma$; group $A$ can enforce $\gamma$ if only Nash equilibria are played (we assume that $\omega_{\mathrm{NE}}$ denotes exactly the set of Nash equilibria in the model).

$\mathcal{L}^{0}_{ATLP}$: $(\mathbf{set\text{-}pl}\ \langle\omega_{\mathrm{NE}},\dots,\omega_{\mathrm{NE}}\rangle)\mathbf{Pl}\,\langle\!\langle A\rangle\!\rangle\gamma$; plausibility terms can be combined. We note the difference to the previous formula, agents are assumed to play a strategy which is part of *some* NE. The resulting strategy profile does not have to be a NE, though.

$\mathcal{L}^{1}_{ATLP}$: $\varphi \equiv (\mathbf{set\text{-}pl}\ \sigma.\exists\sigma_1\varphi'(\sigma,\sigma_1))\mathbf{Pl}\,\langle\!\langle A\rangle\!\rangle\gamma$ where $\varphi'(\sigma,\sigma_1)$ is a formula possibly containing operators $(\mathbf{set\text{-}pl}\ \omega)$ with $\omega \in \mathcal{T}(\Omega\cup\{\sigma,\sigma_1\})$; e.g.

$$\varphi' \equiv (\mathbf{set\text{-}pl}\ \langle\sigma,\dots\sigma,\sigma_1,\omega_{\mathrm{NE}}\rangle)\mathbf{Pl}\,\langle\!\langle A\rangle\!\rangle\gamma'.$$

We have a closer look at the $(\mathbf{set\text{-}pl}\ \cdot)$ operator in $\varphi$. The operator collects all strategies $\sigma$ such that there exists another strategy profile $\sigma_1$ for which $\mathbf{Pl}\,\langle\!\langle A\rangle\!\rangle\gamma'$ holds if all but the last 2 agents play according to $\sigma$, the second to last agent plays according to $\sigma_1$, and the last one according to a fixed strategy out of $\omega_{\mathrm{NE}}$.

$\mathcal{L}^{2}_{ATLP}$: Consider the previous formula $\varphi$ again, but this time $\varphi'(\sigma,\sigma_1)$ can also contain quantification and nesting; e.g.

$$\varphi' \equiv ((\mathbf{set\text{-}pl}\ \langle\sigma,\dots,\sigma,\sigma_1,\omega_{\mathrm{NE}}\rangle)\mathbf{Pl}\,\langle\!\langle B\rangle\!\rangle\gamma') \rightarrow ((\mathbf{set\text{-}pl}\ \sigma'.\exists\sigma_1'\varphi''(\sigma',\sigma_1'))\mathbf{Pl}\,\langle\!\langle A\rangle\!\rangle\gamma)$$

where $\varphi''(\sigma',\sigma_1')$ is a base formula with plausibility terms taken from $\mathcal{T}(\Omega\cup\{\sigma',\sigma_1'\})$.

In the next section we show how the denotation of complex terms is constructed, and how it is plugged into the semantics of **ATLP^base** from Section 6.1.4.

### 6.2.3 Semantics: The Logic ATLP

$\mathcal{L}^{k}_{ATLP}$ does not change the very structure of base formulae, it only extends $\mathcal{L}^{base}_{ATLP}$ by more ornate plausibility terms. Therefore, it seems natural that the plausibility mapping for these terms is of particular interest; the denotation reflects the construction of strategic combinations given in Definition 6.18.

**Definition 6.24 (Extended plausibility mapping $\widehat{[\![\cdot]\!]}$).** *Let* $\mathfrak{M}$ *be* CGSP. *The* extended plausibility mapping $\widehat{[\![\cdot]\!]}_{\mathfrak{M}}$ *with respect to* $[\![\cdot]\!]_{\mathfrak{M}}$ *is defined as follows:*

1. *If* $\omega \in \Omega$ *then* $\widehat{[\![\omega]\!]}^q_{\mathfrak{M}} = [\![\omega]\!]^q_{\mathfrak{M}}$;
2. *If* $\omega = \omega'[A]$ *then* $\widehat{[\![\omega]\!]}^q_{\mathfrak{M}} = \{s \in \Sigma \mid \exists s' \in \widehat{[\![\omega']\!]}^q_{\mathfrak{M}}\ s|_A = s'|_A\}$;
3. *If* $\omega = \langle\omega_1, \dots \omega_k\rangle$ *then* $\widehat{[\![\omega]\!]}^q_{\mathfrak{M}} = \{s \in \Sigma \mid \exists t_1 \in \widehat{[\![\omega_1]\!]}^q_{\mathfrak{M}}, \dots, \exists t_k \in \widehat{[\![\omega_k]\!]}^q_{\mathfrak{M}} \forall i = 1, ..., k\ s|_{a_i} = t_i|_{a_i})\}$;
4. *If* $\omega = \sigma_1.(Q_2\sigma_2)\dots(Q_n\sigma_n)\varphi$ *then*

$$\widehat{[\![\omega]\!]}^q_{\mathfrak{M}} = \{s_1 \in \Sigma \mid Q_2 s_2 \in \Sigma, \dots, Q_n s_n \in \Sigma \quad (\mathfrak{M}^{s_1,\dots,s_n}, q \models \varphi)\},$$

*where* $\mathfrak{M}^{s_1,\dots,s_n}$ *is equal to* $\mathfrak{M}$ *except that we fix* $\Upsilon_{\mathfrak{M}^{s_1,\dots,s_n}} = \Sigma$, $\Omega_{\mathfrak{M}^{s_1,\dots,s_n}} = \Omega_{\mathfrak{M}} \cup \{\sigma_1, \dots, \sigma_n\}$, $[\![\sigma_i]\!]^q_{\mathfrak{M}^{s_1,\dots,s_n}} = \{s_i\}$, *and* $[\![\omega]\!]^q_{\mathfrak{M}^{s_1,\dots,s_n}} = [\![\omega]\!]^q_{\mathfrak{M}}$ *for all* $\omega \neq \sigma_i$, $1 \leq i \leq n$, *and* $q \in Q_{\mathfrak{M}}$. *That is, the denotation of* $\sigma_i$ *in* $\mathfrak{M}^{s_1,\dots,s_n}$ *is set to strategy profile* $s_i$.[3]

Consider, for instance, plausibility term $\sigma_1.\forall\sigma_2\varphi$. The extended plausibility mapping $\widehat{[\![\sigma_1.\forall\sigma_2\varphi]\!]}_q$ collects all strategy profiles $s_1 \in \Sigma$ (referred to by $\sigma_1$) such that *for all* strategy profiles $s_2 \in \Sigma$ (referred to by $\sigma_2$) $\varphi$ is true in model $\mathfrak{M}^{s_1,s_2}$ and state $q \in Q$, i.e. $\mathfrak{M}^{s_1,s_2}, q \models \varphi$ for all $s_2 \in \Sigma$.

*Remark 6.25.* Note that if the language includes a term $\omega_\top$ that refers to *all* strategy profiles, then $x[A]$ can be expressed as $\langle\omega_1, \dots, \omega_k\rangle$, where $\omega_a = x_a$ for $a \in A$, and $\omega_a = \omega_\top$ otherwise. We also observe that in $\mathcal{L}^k_{ATLP}$, $k > 0$, $\omega_\top$ can be expressed as $\sigma.\top$.

In Definition 6.8 we defined the semantics of the base language. Truth of $\mathcal{L}^k_{ATLP}$ formulae is defined in the same way, we only need to replace the previous (simple) plausibility mapping by the extended one in the semantics of plausibility updates.

**Definition 6.26 (Semantics of $\mathcal{L}_{ATLP}$, ATLP).** *The semantics for* $\mathcal{L}_{ATLP}$*-formulae is given as in Definition 6.8 with the extended plausibility mapping* $\widehat{[\![\cdot]\!]}_{\mathfrak{M}}$ *used instead of* $[\![\cdot]\!]_{\mathfrak{M}}$. *I.e., only the semantic clauses for* (***set-pl*** $\omega)\varphi$ *and* (***refn-pl*** $\omega)\varphi$ *change as follows:*

$\mathfrak{M}, q \models_B$ (***set-pl*** $\omega)\varphi$ *iff* $\mathfrak{M}', q \models_B \varphi$ *where the new model* $\mathfrak{M}'$ *is equal to* $\mathfrak{M}$ *but the new set* $\Upsilon_{\mathfrak{M}'}$ *of plausible strategy profiles is set to* $\widehat{[\![\omega]\!]}^q_{\mathfrak{M}}$;

$\mathfrak{M}, q \models_B$ (***refn-pl*** $\omega)\varphi$ *iff* $\mathfrak{M}', q \models_B \varphi$ *where the new model* $\mathfrak{M}'$ *is equal to* $\mathfrak{M}$ *but the new set* $\Upsilon_{\mathfrak{M}'}$ *of plausible strategy profiles is set to* $\Upsilon_{\mathfrak{M}} \cap \widehat{[\![\omega]\!]}^q_{\mathfrak{M}}$.

*We define the logic* **ATLP** *as* $(\mathcal{L}_{ATLP}, \models)$.

[3] It should be emphasised that model $\mathfrak{M}^{s_1,\dots,s_n}$ in which plausibility of profile $s_1$ is evaluated does *not* presuppose any notion of plausibility, i.e., $\Upsilon_{\mathfrak{M}^{s_1,\dots,s_n}} = \Sigma$.

*Remark 6.27.* By slight abuse of notation, we also refer to the extended plausibility mapping with the same symbol as to the simple plausibility mapping, i.e., with $[\![\cdot]\!]$.

We will discuss some important examples of $\mathcal{L}_{ATLP}$-formulae and terms (together with their interpretation) in Sections 6.4.1 and 6.4.2 where **ATLP** characterisations of solution concepts are presented.

## 6.3 Expressiveness of ATLP

In this section, we compare **ATLP** with several related logics and show their formal relationships. To this end, we first define notions that allow to compare expressivity of logical systems. *Embedding* takes place on the level of satisfaction relations ($\models$): Logic $\mathbf{L}_1$ embeds $\mathbf{L}_2$ if models and formulae of $\mathbf{L}_2$ can be simulated in $\mathbf{L}_1$ in a truth-preserving way. *Subsumption* refers to the level of valid sentences: $\mathbf{L}_1$ subsumes $\mathbf{L}_2$ if all the validities of $\mathbf{L}_2$ are validities of $\mathbf{L}_1$ as well.

**Definition 6.28 (Embedding).** Logic $\mathbf{L}_1$ embeds logic $\mathbf{L}_2$ *iff there is a translation tr of* $\mathbf{L}_2$ *formulae into formulae of* $\mathbf{L}_1$*, and a transformation TR of* $\mathbf{L}_2$ *models into models of* $\mathbf{L}_1$*, such that* $\mathfrak{M}, q \models_{\mathbf{L}_2} \varphi$ *iff* $TR(\mathfrak{M}), q \models_{\mathbf{L}_1} tr(\varphi)$ *for each pointed model* $\mathfrak{M}, q$ *and formula* $\varphi$ *of* $\mathbf{L}_2$*.*

Note that the translation of formulae and transformation of models are supposed to be independent. This prevents translation schemes that transform $\mathfrak{M}, q \models \varphi$ in $\mathbf{L}_2$ to $\mathfrak{M}', q \models \top$, and $\mathfrak{M}, q \not\models \varphi$ in $\mathbf{L}_2$ to $\mathfrak{M}', q \not\models \bot$ (with an arbitrary model $\mathfrak{M}'$), that would yield embeddings between any pair of logics.

If not said otherwise all transformations and translation schemes proposed in this section can be computed in polynomial time and incur only a polynomial increase in the size of models and the length of formulae. Thus, we are in fact interested in *polynomial* embeddings of logics in **ATLP**.

**Definition 6.29 (Subsumption).** Logic $\mathbf{L}_1$ subsumes logic $\mathbf{L}_2$ *iff the set of validities of* $\mathbf{L}_1$ *subsumes validities of* $\mathbf{L}_2$*.*

**Proposition 6.30. ATLP** *embeds* **ATL***.*

*Proof.* We use the identity translation of formulae: $tr(\varphi) \equiv \varphi$. As for models, $TR(\mathfrak{M}) = \mathfrak{M}'$ extends $\mathfrak{M}$ with an arbitrary set of plausible strategy profiles $\Upsilon$. It is easy to see that the plausibility assumptions $\Upsilon$ will never be used in the evaluation of $\varphi$ since $\varphi$ includes no **Pl** operators. Thus, the result of the evaluation will be the same as for $\mathfrak{M}, q \models \varphi$. ■

The above reasoning implies also that **ATL** validities hold for all **ATLP** models.

**Corollary 6.31. ATLP** *subsumes* **ATL**.

The relationship of **ATLP** to most other logics can be studied only in the context of embeddings, as they use different modal operators (and thus yield incomparable sets of valid formulae). We begin with embedding **ATLI** [Jamroga et al., 2005] (Section 3.2.5) in **ATLP**. Then, we show that "**CTL** with Plausibility" from [Bulling and Jamroga, 2007a] can be embedded in **ATLP** for a limited (but very natural) class of models. Finally, we propose an embedding of game logic with preferences [van Otterloo et al., 2004; van der Hoek et al., 2004] (cf. Section 3.2.4) which allows to reason about what can happen under *particular* game-theoretical rationality assumptions.

**Proposition 6.32. ATLP** *embeds* **ATLI**.

*Proof.* [Idea] For an **ATLI**-model we construct an CGSP with a plausibility term for each strategic one. Strategically combining plausibility terms allows to restrict only the behaviour of individual agents as in **ATLI**. Then, we have, e.g. $\mathfrak{M}, q \models_{\mathsf{ATLI}} (\mathbf{str}_a \sigma_a)\langle\!\langle A \rangle\!\rangle \Box p$ iff

$$TR(\mathfrak{M}), q \models_{\mathsf{ATLP}} \mathbf{Pl}\,(\mathbf{set\text{-}pl}\ \langle \omega_\top, \ldots, \omega_{\sigma_a}, \ldots, \omega_\top \rangle) \langle\!\langle A \rangle\!\rangle \Box p.$$

The complete proof is given on page 303. ■

**CTLP**, i.e., "**CTL** with Plausibility" [Bulling and Jamroga, 2007a], is an extension of the branching-time logic **CTL** with a similar notion of plausibility as the one we use here. We present the logic in Appendix A.1. The main difference lies in the fact that **CTLP** formulae refer to plausible *paths* rather than strategy profiles. To transform models, we first observe that every transition system $\mathfrak{M}$ can be seen as a concurrent game structure that includes only a single agent $a_1$. Furthermore, we can transform $\mathfrak{M}$ to a CGSP $TR(\mathfrak{M})$ by adding $\Upsilon = \Sigma$ and $\Omega = \emptyset$ (cf. Section 6.1.2).

The main idea of the embedding followed in [Bulling et al., 2009b] to encode (**set-pl** $\gamma$) of **CSLP** as a plausibility term $\sigma.(\mathbf{set\text{-}pl}\ \sigma)\mathbf{Pl}\,\langle\!\langle \emptyset \rangle\!\rangle \gamma$ is flawed. It does not seem possible to use the plausibility operator of **ATLP** to define each set of paths describable by a $\mathcal{L}_{CTL}$-formulae. For illustration we present the following example.

*Example 6.33.* We consider the CGS $\mathfrak{M}$ shown in Figure 6.1. The path formula $\gamma = \Diamond \mathsf{r}$ is true on the $q_0$-paths $(q_0 q_1)^* q_0 (q_2)^\omega$ and $(q_0 q_1)^\omega$ where $^*$ denotes the Kleene star and is understood as for regular expressions. We call this (infinite) set of paths $P$. We claim that this set is *not* representable by *any* set of memoryless strategies. The problem are the paths that visit $q_1$ and $q_2$. Only the path $(q_0 q_1)^\omega$ can be represented by a memoryless strategy. However, then we have that

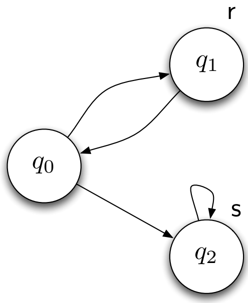



**Fig. 6.1.** A CGS with three states and propositions r and s.

$$\mathfrak{M}, q_0 \models^{\mathsf{ATLP}} (\mathbf{set\text{-}pl}\ \sigma.(\mathbf{set\text{-}pl}\ \sigma)\mathbf{Pl}\,\langle\langle\emptyset\rangle\rangle\gamma)\mathbf{Pl}\,\langle\langle\emptyset\rangle\rangle\Box\neg\mathsf{s}$$

but

$$\mathfrak{M}, q_0 \not\models^{\mathsf{CTLP}} (\mathbf{set\text{-}pl}\ \gamma)\mathbf{Pl}\,\mathsf{A}\neg\mathsf{s}.$$

The proof of the following theorem is based on the result that **CTLP** can be encoded in **CTL**$^+$. We note that the embedding may result in an exponential blow-up of the formula.

**Proposition 6.34.** **ATLP** *embeds* **CTLP** *in the class of transition systems.*

*Proof.* In [Bulling and Jamroga, 2007a] it is shown that every **CTLP** formulae in which plausible paths are described by temporal formulae can be translated to an equivalent **CTL**$^+$-formula. Since **CTL** and **CTL**$^+$ have the same expressive power [Emerson and Halpern, 1985] we can construct an equivalent **CTL**-formula (we note again that this formula may yield an exponential blow-up [Wilke, 1999]). Finally, the **CTL** formula can be embedded in **ATL** and thus also in **ATLP** (Proposition 6.30). ■

**Proposition 6.35.** *If* $\mathbf{P} \neq \mathbf{NP}$ *then* **ATLP** *cannot be polynomially embedded in neither* **ATL**, *nor* **ATLI**, *nor* **CTLP**.

*Proof.* Suppose that any of these logics polynomially embeds **ATLP**. Then, the embedding provides a polynomial reduction of model checking from **ATLP** to that logic. Since model checking of **ATL**, **ATLI**, and **CTLP** can be done in polynomial deterministic time [Alur et al., 2002; Jamroga et al., 2005; Bulling and Jamroga, 2007a], we get that the problem for **ATLP** is in $\mathbf{P}$, too. But model checking **ATLP** is $\mathbf{\Delta_3^P}$-hard already for $\mathcal{L}^{base}_{ATLP}$ (see Section 10.1). ■

There is not much work on logical descriptions of behaviours of agents under rationality assumptions based on game-theoretical solution concepts. In fact, we know only of one such logic for agents with perfect information, which is **GLP** from [van der Hoek et al., 2004] presented in Section 3.2.4. In this logic agents have qualitative preferences (i.e., a propositional formula $\varphi_0$ that they supposedly want to make eventually true). They are assumed to play rationally in the sense that if they have strategies that guarantee $\Diamond\varphi_0$, they can use only those strategies in their play. Interestingly enough, the preference criterion was different in a preliminary version of **GLP** [van Otterloo et al., 2004], where it was based on the notion of Nash equilibrium. We show that **GLP** can be embedded in **ATLP**. One may embed game logics with other preference criteria in an analogous way.

**Proposition 6.36.** **GLP** *can be embedded in* **ATLP**.

*Proof.* [Idea] For the translation of models, we transform game trees of **GLP** to concurrent game structures using the construction from Section 3.2.2, and transform the CGS to a CGSP by taking $\Upsilon = \Sigma$ and $\Omega = \emptyset$. Then, the preference operator $[A : \varphi_0]\psi$ is encoded by setting the plausible strategies to the ones satisfying $\Diamond\varphi_0$. That is, with each subsequent preference operator $[A : \varphi_0]$, only those from the (currently) plausible strategy profiles are selected that are preferred by $a$. The preference is based on the (subgame perfect) enforceability of the outcome $\varphi_0$ at the end of the game: if $\varphi_0$ can be enforced at all, then $a$ prefers strategies that do enforce it. The complete proof is given on page 304. ■

We note that a couple other logics were defined for various solution concepts with respect to incomplete information games [van Otterloo and Roy, 2005; van Otterloo and Jonker, 2004].

*Remark 6.37.* We have presented embeddings of several quite different logics into **ATLP**, which suggests substantial gain in expressive power. Most of them (**ATL**, **ATLI**, and **CTLP**) are embedded already in the lowest levels of the **ATLP** hierarchy (i.e., $\mathbf{ATLP}^{\mathbf{base}}$ or $\mathbf{ATLP}^1$ with no quantifiers). **GLP** formulae with at most $k$ preference operators are embedded in $\mathbf{ATLP}^k$, which is inevitable given their semantics that combines model update and irrevocable strategic quantification (cf. the discussion and the complexity results in [Ågotnes et al., 2007a; Brihaye et al., 2008]).

## 6.4 Solution Concepts Revisited

Solution concepts do not only help to determine the right decision for an agent. Perhaps more importantly, they *constrain* the possible (predicted) responses of the opponents to a proper subset of all the possibilities. For many games the number of all possible outcomes is infinite, although only some of them, often finitely many, *make sense*. We need a notion of rationality (like subgame-perfect Nash equilibrium) to discard the *less sensible* ones, and to determine what should happen had the game been played by ideal players.

While **ATL** is already a logic that incorporates some game theoretical concepts, we claim that extending **ATL** by other useful constructs not only helps us to better understand the classical solution concepts in game theory, but it also paves the way for defining new solution concepts (which we call *general*).

We show in the line with Section 3.3.1 and 3.3.2:

1. That several classical solution concepts for extensive games (Nash equilibria, subgame perfect Nash equilibria, Pareto Optimality), can be characterised already in the language $\mathbf{ATLP}^1$ (Section 6.4.1),
2. That these solution concepts can be also reformulated in a qualitative way, through appropriate formulae of **ATLP** parameterised by **ATL** path formulae (Section 6.4.2).

### 6.4.1 Classical Solution Concepts in $\mathbf{ATLP}^1$

In Section 3.2.2 we have shown how extensive games $\Gamma$ (with a finite set of utilities) can be expressed by CGSs: Each $\Gamma$ can be transformed in a CGS $\mathfrak{M}(\Gamma)$ such that they correspond to each other (in the sense of Definition 3.11).

The following terms rewrite the specification of best response profiles, Nash equilibria, and the specification of subgame-perfect Nash equilibria from Section 3.2.5. Note that the new specifications use only **ATLP** operators.

$$BR_a^T(\sigma) \equiv (\textbf{set-pl}\ \sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}])\mathbf{Pl} \bigwedge_{v \in U} \Big( (\langle\langle a \rangle\rangle T \mathsf{p}_\mathsf{a}^\mathsf{v}) \rightarrow (\textbf{set-pl}\ \sigma)\langle\langle \emptyset \rangle\rangle T \mathsf{p}_\mathsf{a}^\mathsf{v} \Big),$$

$$NE^T(\sigma) \equiv \bigwedge_{a \in \mathbb{A}\mathrm{gt}} BR_a^T(\sigma),$$

$$SPN^T(\sigma) \equiv \langle\langle \emptyset \rangle\rangle \Box NE^T(\sigma).$$

Recalling briefly the ideas behind the above specifications, $BR_a^T(\sigma)$ holds iff $\sigma[a]$ is a best response to $\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]$. That is, after we fix $\mathbb{A}\mathrm{gt} \setminus \{a\}$'s collective strategy to $\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]$, agent $a$ cannot obtain a better temporal pattern of payoffs than by playing $\sigma[a]$. Then, $\sigma$ is a *Nash equilibrium* if each individual strategy $\sigma[a]$ is the best response to the opponent's strategies $\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]$ (cf. [Osborne and Rubinstein, 1994]). The formalisation of a subgame perfect Nash equilibrium is straightforward: We require profile $\sigma$ to be a Nash equilibrium in all reachable states (seen as initial positions of particular subgames).

The following propositions are simple adaptations of the results from Section 3.2.5.

**Proposition 6.38.** *Let $\Gamma$ be an extensive game with a finite set of utilities. Then the following holds:*

1. $s \in [\![\sigma.BR_a^{\Diamond}(\sigma)]\!]^{\emptyset}_{\mathfrak{M}(\Gamma)}$ *iff* $s|_a$ *is a best response for a against* $s|_{\mathbb{A}\mathrm{gt}\setminus\{a\}}$ *in* $\Gamma$.
2. $s \in [\![\sigma.NE^{\Diamond}(\sigma)]\!]^{\emptyset}_{\mathfrak{M}(\Gamma)}$ *iff s is a Nash equilibrium in* $\Gamma$.
3. $s \in [\![\sigma.SPN^{\Diamond}(\sigma)]\!]^{\emptyset}_{\mathfrak{M}(\Gamma)}$ *iff s is a subgame perfect Nash equilibrium in* $\Gamma$.

*Proof.* The proof is done completely analogous to the one given for Proposition 3.15. ■

In Section 3.2.5 we defined a quantitative version of *Pareto optimality* formulated in $\mathcal{L}_{ATLI}$. However, as we pointed out, the $\mathcal{L}_{ATLI}$-formula had exponential length and some counterintuitive implications. Quantification allows to propose a more compact and intuitive specification:

$$PO^T(\sigma) \equiv \quad \forall\sigma' \ \mathbf{Pl}\Big(\bigwedge_{a\in\mathbb{A}\mathrm{gt}}\bigwedge_{v\in U}\big((\textbf{set-pl}\ \sigma')\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}_\mathsf{a}^\mathsf{v} \rightarrow (\textbf{set-pl}\ \sigma)\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}_\mathsf{a}^\mathsf{v}\big) \vee \bigvee_{a\in\mathbb{A}\mathrm{gt}}\bigvee_{v\in U}\big((\textbf{set-pl}\ \sigma)\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}_\mathsf{a}^\mathsf{v} \wedge \neg(\textbf{set-pl}\ \sigma')\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}_\mathsf{a}^\mathsf{v}\big)\Big).$$

This definition of Pareto optimality is more intuitive than the one given in Section 3.2.5 because it does not focus on the temporal evolution of whole payoff profiles, but rather on the interaction between temporal patterns of individual patterns. Although the definition is more intuitive (and thus different from the one of Proposition 3.17) we get the same result. This is, because payoffs are only assigned to leaf nodes if one considers the translation of an extensive form game.

**Proposition 6.39.** *Let $\Gamma$ be an extensive game with a finite set of utilities. Then* $s \in [\![\sigma.PO^{\Diamond}(\sigma)]\!]^{\emptyset}_{\mathfrak{M}(\Gamma)}$ *iff s is Pareto optimal in* $\Gamma$.

*Proof.* "⇒": Let $s \in [\![\sigma.PO^{\Diamond}(\sigma)]\!]^{\emptyset}_{\mathfrak{M}(\Gamma)}$. Then, the formula imposes the following restrictions on $s$. For each payoff profile $\langle v_1, \dots, v_k\rangle$ reachable by some strategy $\sigma'$ in $\Gamma$, it must also be reachable if the agents follow $\sigma$. Or, if for some strategy $\sigma'$ this is not the case; then, there must be some agent $a$ and some payoff $v$ such that the agent can enforce the payoff regarding $\sigma$ but not regarding $\sigma'$. Thus, both profiles are incomparable. The restriction is captured by the restricting to plausible behaviour. Thus, the strategy $\sigma$ is indeed Pareto optimal. Note, that payoffs are only assigned to leaf nodes.

"⇐":The proof for the other direction is done similarly. ■

Let $\langle x^A, y^{\mathbb{A}\mathrm{gt}\setminus A}\rangle$ be a shorthand for the term $\langle z_1, \ldots, z_k\rangle$ with $z_a = x$ for $a \in A$ and $z_a = y$ otherwise. The following specification, formulated as an $\mathcal{L}^1_{ATLP}$-formula, characterises the set of strategy profiles that include undominated strategies for agent $a$:

$$UNDOM^T_a(\sigma) \equiv \quad \forall\sigma_1\forall\sigma_2\exists\sigma_3$$

$$\mathbf{Pl}\Big(\bigwedge_{v\in U}((\mathbf{set\text{-}pl}\ \langle\sigma_1^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}} \rightarrow (\mathbf{set\text{-}pl}\ \langle\sigma^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}})$$

$$\vee \bigvee_{v\in U}((\mathbf{set\text{-}pl}\ \langle\sigma^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}} \wedge \neg(\mathbf{set\text{-}pl}\ \langle\sigma_1^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}})\Big).$$

**Proposition 6.40.** *Let $\Gamma$ be an extensive game with a finite set of utilities. Then, $s \in [\![\sigma.UNDOM^{\Diamond}_a(\sigma)]\!]^{\emptyset}_{\mathfrak{M}(\Gamma)}$ iff $s|_a$ is undominated in $\Gamma$.*

*Proof.* "⇒": Let $s \in [\![\sigma.UNDOM^{\Diamond}_a(\sigma)]\!]^{\emptyset}_{\mathfrak{M}(\Gamma)}$. Assume that $s|_a$ is dominated by some strategy $s'|_a$. That is, $s'|_a$ always yields an outcome for $a$ at least as good as $s|_a$ and strictly better for some profile $s''|_{\mathbb{A}\mathrm{gt}\setminus\{a\}}$. Say, the latter ensures a payoff $v'$. Then, let the denotation of $\sigma_1$ (resp. $\sigma_2$) be such that $[\![\langle\sigma_1^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle]\!] = (s'|_a, s''|_{\mathbb{A}\mathrm{gt}\setminus\{a\}})$. Then, we have that $(\mathbf{set\text{-}pl}\ \langle\sigma_1^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle\Diamond\mathsf{p}^{\mathsf{v}'}_{\mathsf{a}}$ but $\neg(\mathbf{set\text{-}pl}\ \langle\sigma^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle\Diamond\mathsf{p}^{\mathsf{v}'}_{\mathsf{a}})$. Moreover, since $s'|_a$ always yields an outcome for $a$ at least as good as $s|_a$, we also have $(\mathbf{set\text{-}pl}\ \langle\sigma^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle\Diamond\mathsf{p}^{\mathsf{v}}_{\mathsf{a}} \rightarrow (\mathbf{set\text{-}pl}\ \langle\sigma_1^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle\Diamond\mathsf{p}^{\mathsf{v}}_{\mathsf{a}}$ for all payoffs $v$. Contradiction!

"⇐": Suppose $s|_a$ is undominated. Then, for any other profile $s_1|_a$ and $s_2|_{-a}$ the payoff $v$ reachable by $(s_1|_a, s_2|_{-a})$ is also reachable by $(s|_a, s_2|_{-a})$ (this is captured be the left side of the disjunction). Or they are incomparable. That is, there is some payoff reachable by $(s|_a, s_3|_{-a})$ for some strategy $s_3|_{-a}$ of the opponents that is not reachable of $a$ following $s_1|_a$. This is captured by the right side of the conjunction. Thus, the formula is satisfied. ■

### 6.4.2 General Solution Concepts in $\mathcal{L}^1_{ATLP}$

In this section, we return to the idea of *general solution concepts* from Section 3.3.2 and show how qualitative versions of NE, SPN, PO and UNDOM can be captured in **ATLP**. Like for temporalised solution concepts, it turns out that their qualitative counterparts can be already specified in $\mathcal{L}^1_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \emptyset)$. That is, we need only one level of nested plausibility updates (and no "hardwired" plausibility terms) to effectively capture classical notions of rationality and extend them to more general games which we study in this thesis.

We only consider one "winning condition" per agent to represent agents' preferences, but this view can be naturally extended to full preference lists, as in Section 3.5. In what follows, let $\overrightarrow{\eta} = \langle \eta_1, \dots, \eta_k \rangle$ be a vector of $\mathcal{L}_{ATL}$-path formulae. We first define the normal form game corresponding to a CGSP.

**Definition 6.41 (Transforming** CGSP **into normal form game).** *Let* $\mathfrak{M} \in CGSP(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ *and* $q \in Q_{\mathfrak{M}}$*. The* associated NF game $\mathcal{S}(\mathfrak{M}, \overrightarrow{\eta}, q)$ *with respect to* $\overrightarrow{\eta}$ *is given as in Definition 3.19 with* $\mathfrak{M}$ *interpreted as a pure* CGS *by removing* $\Upsilon$ *and* $[\![\cdot]\!]$ *from it.*

Our aim is to define analogues of classical solution concepts (Nash equilibria and such) that are based on explicit "winning conditions" $\eta_i$ instead of numerical payoffs. We can build on our results from the previous section; we only need to replace temporal patterns of payoffs with the formulae $\eta_i$:

$$BR_a^{\overrightarrow{\eta}}(\sigma) \equiv (\textbf{set-pl}\ \sigma[\mathbb{A}\mathrm{gt}\backslash\{a\}])\mathbf{Pl}\left(\langle\!\langle a \rangle\!\rangle \eta_a \rightarrow (\textbf{set-pl}\ \sigma)\langle\!\langle \emptyset \rangle\!\rangle \eta_a\right),$$

$$NE^{\overrightarrow{\eta}}(\sigma) \equiv \bigwedge_{a \in \mathbb{A}\mathrm{gt}} BR_a^{\overrightarrow{\eta}}(\sigma),$$

$$SPN^{\overrightarrow{\eta}}(\sigma) \equiv \langle\!\langle \emptyset \rangle\!\rangle \Box NE^{\overrightarrow{\eta}}(\sigma),$$

$$PO^{\overrightarrow{\eta}}(\sigma) \equiv \forall \sigma' \ \mathbf{Pl}\Big( \bigwedge_{a \in \mathbb{A}\mathrm{gt}} ((\textbf{set-pl}\ \sigma')\langle\!\langle \emptyset \rangle\!\rangle \eta_a \rightarrow (\textbf{set-pl}\ \sigma)\langle\!\langle \emptyset \rangle\!\rangle \eta_a) \vee \bigvee_{a \in \mathbb{A}\mathrm{gt}} ((\textbf{set-pl}\ \sigma)\langle\!\langle \emptyset \rangle\!\rangle \eta_a \wedge \neg(\textbf{set-pl}\ \sigma')\langle\!\langle \emptyset \rangle\!\rangle \eta_a \Big),$$

$$UNDOM_a^{\overrightarrow{\eta}}(\sigma) \equiv \quad \forall \sigma_1 \forall \sigma_2 \exists \sigma_3 \ \mathbf{Pl} \Big(\big((\textbf{set-pl}\ \langle \sigma_1^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\backslash\{a\}} \rangle)\langle\!\langle \emptyset \rangle\!\rangle \eta_a \rightarrow (\textbf{set-pl}\ \langle \sigma^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\backslash\{a\}} \rangle)\langle\!\langle \emptyset \rangle\!\rangle \eta_a\big) \vee \big((\textbf{set-pl}\ \langle \sigma^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\backslash\{a\}} \rangle)\langle\!\langle \emptyset \rangle\!\rangle \eta_a \wedge \neg(\textbf{set-pl}\ \langle \sigma_1^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\backslash\{a\}} \rangle)\langle\!\langle \emptyset \rangle\!\rangle \eta_a\big)\Big).$$

The intuitions of these concepts are the same as in the quantitative case. Note that we did not have to include the big conjunctions/disjunctions over all possible utility values in the case of Pareto optimal and undominated strategies. This is, because the corresponding normal form game can be seen as a game with only *two* possible outcomes per agent.

The following proposition shows that $NE^{\overrightarrow{\eta}}$, $PO^{\overrightarrow{\eta}}$, and $UNDOM^{\overrightarrow{\eta}}$ indeed extend the classical notions of Nash equilibrium, Pareto optimal strategy profile, and undominated strategy.

**Proposition 6.42.**

1. *The set of a's best response strategies in* $\mathcal{S}(\mathfrak{M}, \overrightarrow{\eta}, q)$ *is given by*

$$[\![\sigma.BR_a^{\overrightarrow{\eta}}(\sigma)]\!]_{\mathfrak{M}}^q.$$

*2. The set of Nash equilibrium strategies in* $\mathcal{S}(\mathfrak{M}, \overrightarrow{\eta}, q)$ *is given by*

$$[\![\sigma.NE^{\overrightarrow{\eta}}(\sigma)]\!]^q_{\mathfrak{M}}.$$

*3. The set of Pareto optimal strategies in* $\mathcal{S}(\mathfrak{M}, \overrightarrow{\eta}, q)$ *is given by*

$$[\![\sigma.PO^{\overrightarrow{\eta}}(\sigma)]\!]^q_{\mathfrak{M}}.$$

*4. The set of a's undominated strategies in* $\mathcal{S}(\mathfrak{M}, \overrightarrow{\eta}, q)$ *is given by* $([\![\sigma.UNDOM^{\overrightarrow{\eta}}_a(\sigma)]\!]^q_{\mathfrak{M}})|_a$.

*Proof.* Let $\mathcal{S} = \mathcal{S}(\mathfrak{M}, \overrightarrow{\eta}, q)$.

1. "$\subseteq$": Suppose $s_a$ is a best response to $s_{-a}$ in $\mathcal{S}$. Let $\sigma$ be the strategic variable with denotation $s = (s_a, s_{-a})$. Then, if $\mu_a(s) = 0$ there is no other strategy $s'_a$ of $a$ such that $\mu_a(s'_a, s_{-a}) = 1$. Now, assume that $(\textbf{set-pl}\ \sigma[\mathbb{A}\text{gt}\backslash\{a\}])\textbf{Pl}\ \langle\!\langle a\rangle\!\rangle\eta_a$ holds in $\mathfrak{M}, q$. Then, there is a strategy $s'_a$ of $a$ such that $\eta_a$ holds along all paths from $out(q, (s'_a, s_{-a}))$; hence, $\mu_a(s'_a, s_{-a}) = 1$. Now, suppose that $s = (s_a, s_{-a}) \notin [\![\sigma.\widehat{BR^{\boldsymbol{\eta}}_a(\sigma)}]\!]^q_{\mathfrak{M}}$; i.e., that $(\textbf{set-pl}\ \sigma)\langle\!\langle\emptyset\rangle\!\rangle\eta_a$ does not hold in $\mathfrak{M}, q$. Then, there is a path in $out(q, s)$ along which $\eta_a$ is false and thus $\mu_a(s) = 0$. Contradiction!
   "$\supseteq$": Suppose $s \in [\![\sigma.\widehat{BR^{\boldsymbol{\eta}}_a(\sigma)}]\!]^q_{\mathfrak{M}}$ with $[\![\sigma]\!]^q_{\mathfrak{M}} = s$. That is, $BR^{\eta}_a(\sigma)$ is true in $\mathfrak{M}, q$. Then, following the same reasoning as above we have that if $\mu_a(s) = 0$; then, there is no other strategy $s'_a$ of $a$ such that $\mu_a(s'_a, s_{-a}) = 1$. I.e. $s_a$ is a best response to $s_{-a}$.
2. Follows from 1 and the fact that $s$ is a NE iff $s_a$ is a best response to $s_{-a}$ for each agent $a$.
3. "$\subseteq$": Let $s$ be Pareto optimal in $\mathcal{S}$. That is, there is no profile $s'$ such that for all agents $a$, $\mu_a(s') \geq \mu_a(s)$ and for some agent $a$, $\mu_a(s') > \mu_a(s)$. We show that $\mathfrak{M}, q \models PO^{\boldsymbol{\eta}}(\sigma)$. For the sake of contradiction assume the contrary; that is, there is $s'$ such that $\mathfrak{M}^{s'}, q \models \textbf{Pl}\ \big(\bigvee_{a\in\mathbb{A}\text{gt}}((\textbf{set-pl}\ \sigma')\langle\!\langle\emptyset\rangle\!\rangle\eta_a \wedge \neg(\textbf{set-pl}\ \sigma)\langle\!\langle\emptyset\rangle\!\rangle\eta_a) \wedge \bigwedge_{a\in\mathbb{A}\text{gt}}((\textbf{set-pl}\ \sigma)\langle\!\langle\emptyset\rangle\!\rangle\eta_a \rightarrow (\textbf{set-pl}\ \sigma')\langle\!\langle\emptyset\rangle\!\rangle\eta_a)\big)$. We use $s'$ to denote $\sigma'$ for which the formula evaluates true. According to the left-hand site of the outermost (wrt. infix notation) conjunction, there has to be an agent $a'$ such that $\eta_{a'}$ is achievable with respect to $s'$ but not with respect to $s$. From the right-hand side of the conjunction, we learn that the profile $s'$ is at least as good as $s$ (every payoff achievable following $s$ is also achievable following $s'$). However, this means that $s$ is not Pareto optimal. Contradiction!
   "$\supseteq$": This part follows the same reasoning as the other direction.
4. "$\subseteq$": Suppose $s_a$ is undominated in $\mathcal{S}(\mathfrak{M}, \overrightarrow{\eta}, q)$. That is, there is no other strategy $s'_a$ such that for all $s_{-a}$, $\mu_a(s'_a, s_{-a}) \geq \mu_a(s_a, s_{-a})$ and for some $s'_{-a}$, $\mu_a(s'_a, s'_{-a}) > \mu_a(s_a, s'_{-a})$. Suppose that $s_a \notin ([\![\sigma.UNDOM^{\overrightarrow{\eta}}_a(\sigma)]\!]^q_{\mathfrak{M}})|_a$.

Hence, there are profiles $\sigma_1, \sigma_2$ such that for all profiles $\sigma_3$ we have that $(\textbf{set-pl}\ \langle \sigma_1^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle\eta_a$ and $\neg(\textbf{set-pl}\ \langle \sigma^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle\eta_a$ and $(\textbf{set-pl}\ \langle \sigma^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle\eta_a \rightarrow (\textbf{set-pl}\ \langle \sigma_1^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\langle\emptyset\rangle\rangle\eta_a$. The first part says that $\sigma_1[a]$ is strictly better than $\sigma[a]$ regarding $\sigma_2[\mathbb{A}\mathrm{gt}\setminus\{a\}]$ (i.e. yields a better payoff). The second part expresses that $\sigma_1[a]$ is at least as good as $\sigma[a]$ against all responses $\sigma_3[\mathbb{A}\mathrm{gt}\setminus\{a\}]$ of the opponents. Contradiction to the assumption that $s_a$ is undominated.
"$\supseteq$": This part follows the same reasoning as the other direction.

■

Subgame perfect Nash equilibria are related to normal form games in the following way.

**Proposition 6.43.** *Let $Q'$ be the set of states reachable from $q$ in $\mathfrak{M}$. Then,* $[\![\sigma.SPN^\eta(\sigma)]\!]^q_{\mathfrak{M}} = \bigcap_{q\in Q'}[\![\sigma.NE^\eta(\sigma)]\!]^q_{\mathfrak{M}}$.

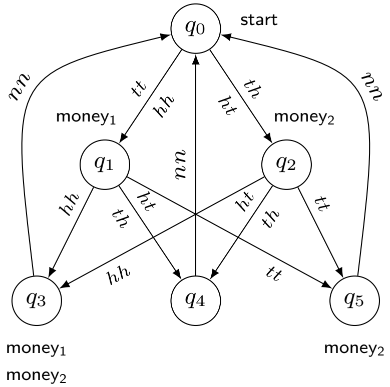
*Proof.* We have that $s \in [\![\sigma.\widehat{SPN^{\boldsymbol{\eta}}}(\sigma)]\!]^q_{\mathfrak{M}}$ iff for all paths $\lambda \in out(q, s_\emptyset)$ and all $i \in \mathbb{N}_0$, $\mathfrak{M}, \lambda[i,\infty] \models NE^{\boldsymbol{\eta}}(\sigma)$ iff $\forall q' \in Q'$, $\mathfrak{M}, q' \models NE^{\boldsymbol{\eta}}(\sigma)$ iff $\forall q' \in Q'$, $s \in [\![\sigma.\widehat{NE^{\boldsymbol{\eta}}}(\sigma)]\!]^{q'}_{\mathfrak{M}}$ iff $s \in \bigcap_{q'\in Q'} [\![\sigma.\widehat{NE^{\boldsymbol{\eta}}}(\sigma)]\!]^{q'}_{\mathfrak{M}}$. ■

*Example 6.44 (Extended matching pennies).* In Figure 6.2 we consider a slightly more complex version of the *asymmetric matching pennies game* presented in Figure 3.5. The new game consists of two phases (played *ad infinitum*). First, player 1 wins some money if the sides of the pennies match, otherwise the money goes to player 2. In the second phase, both win a prize if both show heads; if they both show tails, only player 2 wins. If they show different sides, nobody wins.

We denote particular strategies as $s_{\alpha_1\alpha_2\alpha_3}$, where $\alpha_1$ is the action played at state $q_0$, $\alpha_2$ is the action played at state $q_1$, and $\alpha_3$ is played at $q_2$. We note that every combination of strategies (i.e., every strategy profile) determines a single temporal path. For example, if agent 1 plays $s_{htt}$ and agent 2 plays $s_{ttt}$, then they ensure the (infinite) temporal path $q_0q_2q_5(q_0q_2q_5)^\omega$.

Let us additionally assume that the winning conditions are:
$\eta_1 \equiv \Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1)$ for player 1 and $\eta_2 \equiv \Diamond\mathsf{money}_2$ for player 2. That is, agent 1 is only happy if it gets money all the time (whenever possible). Agent 2 is more minimalistic: It is sufficient for it to win money once, sometime in the future. So, for instance, the play that results from strategy profile $\langle s_{htt}, s_{ttt}\rangle$ satisfies the second player's preferences, but not the first player ones. This way, it is easy to construct a table of binary payoffs that indicates which strategy profiles are "winning" for whom, shown in the table in Figure 6.2B. Now, we can for instance observe that the profile $\langle s_{htt}, s_{ttt}\rangle$ is a Nash equilibrium (player 1 cannot make itself happy by unilaterally changing its strategy), but it

(A)




(B)

| $\eta_1 \backslash \eta_2$ | $s_{hhh}$ | $s_{hht}$ | $s_{hth}$ | $s_{htt}$ | $s_{thh}$ | $s_{tht}$ | $s_{tth}$ | $s_{ttt}$ |
|---|---|---|---|---|---|---|---|---|
| $s_{hhh}$ | **1,1** | **1,1** | 0,0 | 0,0 | 0,1 | 0,1 | 0,1 | 0,1 |
| $s_{hht}$ | **1,1** | **1,1** | 0,0 | 0,0 | 0,1 | 0,1 | 0,1 | 0,1 |
| $s_{hth}$ | 0,0 | 0,0 | 0,1 | 0,1 | 0,1 | 0,1 | 0,1 | 0,1 |
| $s_{htt}$ | 0,0 | 0,0 | 0,1 | 0,1 | 0,1 | 0,1 | 0,1 | 0,1 |
| $s_{thh}$ | 0,1 | 0,1 | 0,1 | 0,1 | **1,1** | **1,1** | 0,0 | 0,0 |
| $s_{tht}$ | 0,1 | 0,1 | 0,1 | 0,1 | **1,1** | **1,1** | 0,0 | 0,0 |
| $s_{tth}$ | 0,1 | 0,1 | 0,1 | 0,1 | 0,0 | 0,0 | 0,1 | 0,1 |
| $s_{ttt}$ | 0,1 | 0,1 | 0,1 | 0,1 | 0,0 | 0,0 | 0,1 | 0,1 |

**Fig. 6.2.** "Extended matching pennies": (A) CGS $\mathfrak{M}_3$; again, action profile $xy$ refers to action $x$ played by player 1 and action $y$ played by 2. (B) Strategies and their outcomes for $\eta_1 \equiv \Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1)$, $\eta_2 \equiv \Diamond\mathsf{money}_2$. Pareto optimal profiles are indicated with bold font, Nash equilibria with grey background.

is not Pareto optimal ($\langle s_{hhx}, s_{hhy}\rangle$ and $\langle s_{thx}, s_{thy}\rangle$ yield strictly better payoff profiles for $x, y \in \{t, h\}$). As before, the CGS $\mathfrak{M}_3$ in Figure 6.2A can be seen as a CGSP by adding $\Upsilon = \Sigma$ and $\Omega = \emptyset$. Now, we have that:

- $[\![\sigma.NE^{\eta_1,\eta_2}(\sigma)]\!]^{q_0}_{\mathfrak{M}_3} = \{\langle s_{hhx}, s_{hhy}\rangle, \langle s_{hhx}, s_{tty}\rangle, \langle s_{htx}, s_{hty}\rangle, \langle s_{htx}, s_{tty}\rangle, \langle s_{thx}, s_{hty}\rangle, \langle s_{thx}, s_{thy}\rangle, \langle s_{ttx}, s_{hty}\rangle, \langle s_{ttx}, s_{tty}\rangle \mid x, y \in \{h, t\}\}$, and
- $[\![\sigma.PO^{\eta_1,\eta_2}(\sigma)]\!]^{q_0}_{\mathfrak{M}_3} = \{\langle s_{hh}, s_{hh}\rangle, \langle s_{th}, s_{th}\rangle \mid x, y \in \{h, t\}\}$.

Suppose that agent 1 wants money always, and 2 wants money eventually, and only Pareto optimal Nash equilibria are played. Then, agent 1 is bound to get money at the beginning of each round of the game. Formally:

$$\mathfrak{M}_3, q_0 \models (\textbf{set-pl}\ \sigma.NE^{\eta_1,\eta_2}(\sigma))$$
$$(\textbf{refn-pl}\ \sigma.PO^{\eta_1,\eta_2}(\sigma))\textbf{Pl}\,(\mathsf{start} \rightarrow \langle\!\langle\emptyset\rangle\!\rangle \bigcirc \mathsf{money}_1).$$

In **ATLP**, we can also describe relationships between different solution concepts in a CGS. For example, in the "extended matching pennies" game, all Pareto optimal profiles happen to be a Nash equilibrium, which is equivalent to the following formula:

$$(\textbf{set-pl}\ \sigma.PO^{\eta_1,\eta_2}(\sigma))(\textbf{refn-pl}\ \sigma.\neg NE^{\eta_1,\eta_2}(\sigma))\mathbf{Pl}\,\neg\langle\langle \mathbb{A}\mathrm{gt}\rangle\rangle \bigcirc \top,$$

and the formula does indeed hold in $\mathfrak{M}_3, q_0$.

## 6.5 Abilities of Rational Coalitions: CoalATL

In the previous sections we have recalled ATLs and have shown that these temporal logics can be used for reasoning about the abilities of agents. The logic **ATLP** can be used to fix some notions of plausibility/rationality so that choices of specified groups of agents are restricted. In all these logics the key construct has the form $\langle\langle A\rangle\rangle\varphi$, which expresses that coalition $A$ of agents can *enforce* formula $\varphi$. Under a model theoretic viewpoint, $\langle\langle A\rangle\rangle\varphi$ holds whenever the agents in $A$ have a winning strategy for ensuring that $\varphi$ holds (independently of the behaviour of $A$'s opponents). However, this operator accounts only for the *theoretical existence* of such a strategy, not taking into account whether the coalition $A$ can be actually formed. Indeed, in order to join a coalition, agents usually require some kind of *incentive* (e.g. sharing common goals, getting rewards, etc.), since usually forming a coalition does not come for free (fees have to be paid, communication costs may occur, etc.). Consequently, several possible coalition structures among agents may arise, from which the best ones should be adopted according to some rationally justifiable procedure.

In this section we present an argumentative approach to extend **ATL** for modelling coalitions. We provide a formal extension of **ATL**, **CoalATL** (*Coalitional* **ATL** ), by including a new construct $\langle\!|A|\!\rangle\varphi$ which denotes that *the group $A$ of agents is able to build a coalition $B$, $A\cap B \neq \emptyset$ provided that $A \neq \emptyset$, such that $B$ can enforce $\varphi$*. That is, it is assumed that agents in $A$ work together and try to form a coalition $B$.

The main inspiration for our work is the argument-based model for reasoning about coalition structures proposed by Amgoud [Amgoud, 2005a]. Indeed, our notion of coalitional framework (Def. 4.39) is based on the notion of framework for generating coalition structures (FCS) presented in Amgoud's paper. However, in contrast with Amgoud's proposal, our work is concerned with extending **ATL** by argumentation in order to model coalition formation.

Previous research by Hattori *et al.* [Hattori et al., 2001] has also addressed the problem of argument-based coalition formation, but from a different perspective than ours. In [Hattori et al., 2001] the authors propose an argumentation-based negotiation method for coalition formation which combines a logical framework and an argument evaluation mechanism. The resulting system involves several user agents and a mediator agent. During the negotiation, the mediator agent encourages appropriate user agents to join in a coalition in order to facilitate reaching an agreement. User agents advance proposals using a part of the user's valuations in order to reflect the user's preferences in an agreement. This approach differs greatly from our proposal, as we are not concerned with the negotiation process among agents, and our focus is on modelling coalitions within an extension of an expressive strategic logic, where coalition formation is part of the logical language.

Modelling argument-based reasoning with bounded rationality has also been the focus of previous research. In [Rovatsos et al., 2005] the authors propose the use of a framework for argument-based negotiation, which allows for a strategic and adaptive communication to achieve private goals within the limits of bounded rationality in open argumentation communities. In contrast with our approach, the focus here is not on extending a particular logic for reasoning about coalitions, as in our case. Recent research in formalising coalition formation has been oriented towards adding more expressivity to Pauly's coalition logic [Pauly, 2002]. E.g. in [Ågotnes et al., 2007b], the authors define *Quantified Coalition Logic*, extending coalition logic with a limited but useful form of quantification to express properties such as *"there exists a coalition $C$ satisfying property $P$ such that $C$ can achieve $\varphi$"*. In [Borgo, 2007], a semantic translation from coalition logic to a fragment of an action logic is defined, connecting the notions of coalition power and the actions of the agents. However, in none of these cases argumentation is used to model the notion of coalition formation as done in this thesis.

It must be noted that the adequate formalisation of preferences has deserved considerable attention within the argumentation community. In preference-based argumentation theory, an argument may be preferred to another one when, for example, it is more specific, its beliefs have a higher probability or certainty, or it promotes a higher value. Recent work by Kaci *et al.* [Kaci and van der Torre, 2008; Kaci et al., 2007] has provided interesting contributions in this direction, including default reasoning abilities about the preferences over the arguments, as well as an algorithm to derive the most likely preference order.

### 6.5.1 Rational Coalition Formation

During the last decade, argumentation frameworks [Prakken and Vreeswijk, 2002; Chesñevar et al., 2000] have evolved as a successful approach to formalise common-sense reasoning and decision making in multiagent systems (MASs). Application areas include issues such as joint deliberation, persuasion, negotiation, knowledge distribution and conflict resolution (e.g. [Tang and Parsons, 2005; Rahwan and Amgoud, 2006; Rahwan et al., 2007; Brena et al., 2007; Karunatillake et al., 2006]), among many others. Particularly, recent research by Leila Amgoud [Amgoud, 2005a,b] has shown that argumentation provides a sound setting to model *reasoning about coalition formation* in MASs. The approach is based on using conflict and preference relationships among coalitions to determine which coalitions should be adopted by the agents. This is done according to a particular argumentation semantics. The work is presented in Section 4.2 The actual computation of the coalition is modelled in terms of a given argumentation semantics [Dung, 1995] in the context of coalition formation [Amgoud, 2005a].

The formation process is quite abstract. As mentioned above, agents usually require some kind of *incentive* (e.g. sharing common goals, getting rewards, etc.), since usually forming a coalition does not come for free (fees have to be paid, communication costs may occur, etc.). Thus, in a second step we consider *goals* as the main motivation to join coalitions. Agents should only work together if the group is conflict-free (i.e. acceptance according to the coalition formation framework just discussed) and each agent inside the coalition should somewhat benefit from the participation. In Section 6.6 we make this idea formal and enrich **CoalATL** with goals. We address the question *why* agents should cooperate. Goals refer to agents' subjective incentives to join coalitions.

### 6.5.2 The Language $\mathcal{L}_{CoalATL}$

In this section we combine *argumentation for coalition formation* from Section 4.2 and $\mathcal{L}_{ATL}$ and introduce the language $\mathcal{L}_{CoalATL}$. The latter extends $\mathcal{L}_{ATL}$ by new operators $\langle\!| A |\!\rangle$ for each subset $A \subseteq \mathbb{A}\mathrm{gt}$ of agents. These new modalities combine, or rather integrate, coalition formation into the original cooperation modalities $\langle\!\langle A \rangle\!\rangle$. The intended reading of $\langle\!| A |\!\rangle\varphi$ is that the group $A$ of agents *is able to form a coalition* $B \subseteq \mathbb{A}\mathrm{gt}$ *such that some agents of* $A$ *are also members of* $B$, if $A \neq \emptyset$ then $A \cap B \neq \emptyset$, and $B$ *can enforce* $\varphi$.

Our main motivation for this logic is to make it possible to reason about the ability of building coalition structures, and not only about an *a priori* specified group of agents (as it is the case for $\langle\!\langle A \rangle\!\rangle\varphi$). The new modality $\langle\!| A |\!\rangle$ provides a rather subjective view to the agents in $A$ and their power to create a group $B$ which in turn is used to reason about the ability to enforce a given property.

The language of $\mathcal{L}_{CoalATL}$ is defined as follows.

**Definition 6.45 ($\mathcal{L}_{CoalATL}$).** *The language* $\mathcal{L}_{CoalATL}(\Pi, \mathbb{A}\mathrm{gt})$ *is defined by the following grammar:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A \rangle\!\rangle \bigcirc \varphi \mid \langle\!\langle A \rangle\!\rangle \Box\varphi \mid \langle\!\langle A \rangle\!\rangle \varphi\mathcal{U}\varphi \mid \langle\!| A |\!\rangle \bigcirc \varphi \mid \langle\!| A |\!\rangle \Box\varphi \mid \langle\!| A |\!\rangle \varphi\mathcal{U}\varphi$$

*where* $A \subseteq \mathbb{A}\mathrm{gt}$ *and* $\mathsf{p} \in \Pi$.

### 6.5.3 Semantics: The Logic CoalATL

We extend concurrent game structures by *coalitional frameworks* and the *argumentative semantics* from Section 4.2, Definitions 4.39 and 4.44. A coalitional framework is assigned to each state of the model capturing the current "conflicts" among agents. In doing so, we allow that conflicts change over time, being thus *state dependent.* Moreover, we assume that coalitional frameworks depend on groups of agents. Two initial groups of agents may have different skills to form coalitions. Consider for instance the following example.

*Example 6.46.* Imagine the two agents $a_1$ and $a_2$ are not able (because they do not have the money) to convince $a_3$ and $a_4$ to join. But $a_1$, $a_2$ and $a_3$ together have the money and all four can enforce a property $\varphi$. So $\{a_1, a_2\}$ are not able to build a greater coalition to enforce $\varphi$; but $\{a_1, a_2, a_3\}$ are. So we are not looking at coalitions per se, but how they evolve from others.

We assume that the argumentative semantics is the same for all states.

**Definition 6.47** (CGM). *A* coalitional game model *(*CGM*) is given by a tuple*

$$\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \zeta, \mathfrak{sem} \rangle$$

*where* $\langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o \rangle$ *is a* CGS, $\zeta : \mathcal{P}(\mathbb{A}\mathrm{gt}) \to (Q \to \mathbb{CF}(\mathbb{A}\mathrm{gt}))$ *is a function which assigns a coalitional framework over* $\mathbb{A}\mathrm{gt}$ *to each state of the model subjective to a given group of agents, and* $\mathfrak{sem}$ *is an (argumentative) semantics as defined in Definition 4.44. The set of all such models is given by* $\mathbb{M}(Q, \mathbb{A}\mathrm{gt}, \Pi, \mathfrak{sem}, \zeta)$.

A model provides an argumentation semantics $\mathfrak{sem}$ which assigns all formable coalitions to a given coalitional framework. As argued before we require from a valid coalition that it is not only justified by the argumentation semantics but that it is also not disjunct with the predetermined starting coalition. This leads to the notion *valid coalition.*

**Definition 6.48 (Valid coalition).** *Let* $A$, $B \subseteq \mathbb{A}\mathrm{gt}$ *be groups of agents,* $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \zeta, \mathfrak{sem} \rangle$ *be a* CGM *and* $q \in Q$.

*We say that* $B$ *is a* valid coalition *with respect to* $A$, $q$, *and* $\mathfrak{M}$ *whenever* $B \in \mathfrak{sem}(\zeta(A)(q)))$ *and if* $A \neq \emptyset$ *then* $A \cap B \neq \emptyset$. *Furthermore, we use* $VC_{\mathfrak{M}}(A, q)$ *to denote the set of all valid coalitions regarding* $A$, $q$, *and* $\mathfrak{M}$. *The subscript* $\mathfrak{M}$ *is omitted if clear from context.*

*Remark 6.49.* In [Bulling et al., 2008] we assume that the members of the initial group $A$ work together, whatever the reasons might be. So group $A$ was added to the semantics. This ensured that agents in $A$ can enforce $\psi$ on their own, if they are able to do so. Even if $A$ is *not* accepted originally by the argumentation semantics, i.e. $A \notin \mathfrak{sem}(\zeta(A)(q))$. Here, we do not require this condition. As pointed out in [Bulling and Dix, 2008] the "old" semantics is just a special case of this new one: The operator from [Bulling et al., 2008] can be defined as $\langle\!| A |\!\rangle\gamma \vee \langle\!\langle A \rangle\!\rangle\gamma$.

Moreover, we changed the condition that the predefined group given in the coalitional operator must be a subset of the formed coalition, $A \subseteq B$, to the requirement that some member of the initial coalition (if $A \neq \emptyset$) should be in the new one, $A \cap B \neq \emptyset$. Both definitions make sense in different scenarios; however, the new one seems to be more generic.

The semantics of the new modality is given by

**Definition 6.50 (CoalATL semantics).** *Let a* CGM $\mathfrak{M}$*, a group of agents* $A \subseteq \mathbb{A}\text{gt}$*, and* $q \in Q$ *be given. The semantics of* **CoalATL** *extends the one of* **ATL***, given in Definition 2.22, by the following rule where* $\langle\!| A |\!\rangle\psi \in \mathcal{L}_{CoalATL}(\mathbb{A}\text{gt}, \Pi)$*:*

$\mathfrak{M}, q \models \langle\!| A |\!\rangle\psi$ *iff there is a coalition* $B \in VC(A, q)$ *such that* $\mathfrak{M}, q \models \langle\!\langle B \rangle\!\rangle\psi$.

*Remark 6.51 (Different Semantics,* $\models_{\mathfrak{sem}}$*).* We have just defined a whole class of semantic rules for modality $\langle\!| \cdot |\!\rangle$. The actual instantiation of the semantics $\mathfrak{sem}$, for example $\mathfrak{sem}_{\text{stable}}$, $\mathfrak{sem}_{\text{pref}}$, and $\mathfrak{sem}_{\text{cs}}$ defined in Section 4.2, affects the semantics of the cooperation modality.

For the sake of readability, we sometimes annotate the satisfaction relation $\models$ with the presently used argumentation semantics. That is, given a CGM $\mathfrak{M}$ with an argumentation semantics $\mathfrak{sem}$ we write $\models_{\mathfrak{sem}}$ instead of $\models$.

The underlying idea of the semantic definition of $\langle\!| A |\!\rangle\psi$ is as follows. A given (initial) group of agents $A \subseteq \mathbb{A}\text{gt}$ is able to form a *valid coalition* $B$ (where $A$ and $B$ must not be disjoint if $A \neq \emptyset$), with respect to a given coalitional framework $\mathcal{CF}$ and a particular semantics $\mathfrak{sem}$, such that $B$ can enforce $\psi$.

*Remark 6.52.* Similarly to the alternatives to our definition of valid coalitions there are other sensible semantics for **CoalATL**. The semantics we presented here is not particularly dependent on time; i.e., except from the selection of a valid coalition $B$ at the initial state there is no further interaction between time and coalition formation. We have chosen this simplistic definition to present our main idea–the connection of **ATL** and coalition formation by means of argumentation–as clear as possible.

In the semantics presented in Definition 6.50 a valid coalition is initially formed and kept until the property is fulfilled. For instance, consider formula $\langle\!\lvert A \rvert\!\rangle \Box\varphi$. The formula is true in $q$ if a valid coalition $B$ *in* $q$ can be formed such that it can ensure $\Box\varphi$. On might strengthen the scenario and require that $B$ must be valid in *each* state on the path $\lambda$ satisfying $\varphi$. Formally, the semantics could be given as follows: $q \models \langle\!\lvert A \rvert\!\rangle \Box\varphi$ if, and only if, $q \models \varphi$ and there is a coalition $B \in \mathrm{VC}(q, A)$ and a common strategy $s_B \in \Sigma_B$ such that for all paths $\lambda \in out(q, s_B)$ and for all $i \in \mathbb{N}_0$ it holds that $\lambda[i] \models \varphi$ *and* $B \in \mathrm{VC}(\lambda[i], A)$. The last part specifies that $B$ must be a valid coalition in each state $q_i = \lambda[i]$ of $\lambda$.

In the semantics just presented the formed coalition $B$ must persist over time until $\varphi$ is enforced. One can go one step further. Instead of keeping the same coalition $B$ it can also be sensible to consider "new" valid coalitions in each time step (wrt. $A$), possibly distinct from $B$. This leads to some kind of fixed-point definition. At first, $B$ must be a valid coalition in state $q$ leading to a state in which $\varphi$ is fulfilled and in which another valid coalition (wrt. $A$ and the new state) exists which in turn can ensure to enter a state in which, again, there is another valid coalition and so on.

**Proposition 6.53.** *Let* $A \subseteq \mathbb{A}\mathrm{gt}$ *and* $\langle\!\lvert A \rvert\!\rangle \psi \in \mathcal{L}_{CoalATL}(\mathbb{A}\mathrm{gt}, \Pi)$. *Then* $\langle\!\lvert A \rvert\!\rangle \psi \rightarrow \bigvee_{B \in \mathcal{P}(\mathbb{A}\mathrm{gt})} \langle\!\langle B \rangle\!\rangle \psi$ *is valid in the class of* CGMs.

*Proof.* Suppose $\mathfrak{M}, q \models \langle\!\lvert A \rvert\!\rangle \psi$. Then, there is a valid coalition $B$ such that $B$ can enforce $\psi$. Since $B$ is valid we particularly have that $B \subseteq \mathbb{A}\mathrm{gt}$. ∎

Compared to **ATL**, a $\mathcal{L}_{CoalATL}$-formula like $\langle\!\lvert A \rvert\!\rangle \varphi$ does *not* refer to the ability of $A$ to enforce $\varphi$, but rather to the ability of $A$ to *constitute* a coalition $B$, such that $A \cap B \neq \emptyset$ provided $A \neq \emptyset$, and then, in a second step, to the ability of $B$ to enforce $\varphi$. Thus, two different notions of ability are captured in these new modalities. For instance, $\langle\!\langle A \rangle\!\rangle \psi \wedge \neg \langle\!\lvert \emptyset \rvert\!\rangle \psi$ expresses that group $A$ of agents can enforce $\psi$, but there is no *reasonable* coalition which can enforce $\psi$ (particularly not $A$, although they possess the theoretical power to do so).

*Example 6.54.* There are three agents $a_1$, $a_2$, and $a_3$ which prefer different outcomes. Agent $a_1$ (resp. $a_2$, $a_3$) desires to get outcome $\mathsf{r}$ (resp. $\mathsf{s}$, $\mathsf{t}$). One may assume that all outcomes are distinct; for instance, $a_1$ is not satisfied with an outcome $x$ whenever $x \neq \mathsf{r}$. Each agent can choose to perform action $\alpha$ or $\beta$. Action profiles and their outcomes are shown in Figure 6.3. The $\star$ is used as a placeholder for any of the two actions, i.e. $\star \in \{\alpha, \beta\}$. For instance, the profile $(\beta, \beta, \star)$ leads to state $q_3$ whenever agent $a_1$ and $a_2$ perform action $\beta$ and $a_3$ either does $\alpha$ or $\beta$.

According to the scenario depicted in the figure, $a_1$ and $a_2$ cannot commonly achieve their goals. The same holds for $a_1$ and $a_3$. On the other hand, there exists a situation, $q_1$, in which both agents $a_2$ and $a_3$ are satisfied. One

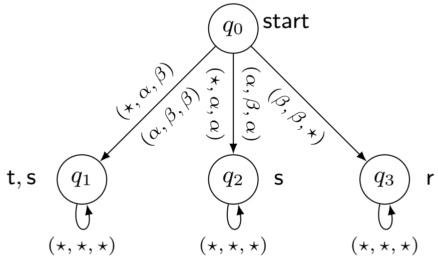



**Fig. 6.3.** A simple CGS defined in Example 6.54.

can formalise the situation as the coalitional game $\mathcal{CF} = (\mathfrak{C}, \mathcal{A}, \prec)$ given in Example 4.49(b), that is, $\mathfrak{C} = \mathbb{A}\mathrm{gt}$, $\mathcal{A} = \{(a_1, a_2), (a_1, a_3), (a_2, a_1), (a_2, a_3), (a_3, a_1)\}$ and $a_2 \prec a_3$.

We formalise the example as the CGM $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \zeta, \mathfrak{sem} \rangle$ where $\mathbb{A}\mathrm{gt} = \{a_1, a_2, a_3\}, Q = \{q_0, q_1, q_2, q_3\}$, $\Pi = \{\mathsf{r}, \mathsf{s}, \mathsf{t}\}$, and $\zeta(A)(q) = \mathcal{CF}$ for all states $q \in Q$ and groups $A \subseteq \mathbb{A}\mathrm{gt}$. Transitions and the state labelling can be seen in Figure 6.3. Furthermore, we do not specify a concrete semantics $\mathfrak{sem}$ yet, and rather adjust it in the remainder of the example.

We can use pure $\mathcal{L}_{ATL}$-formulas, i.e. formulas not containing the new modalities $\langle\!\![ \cdot ]\!\!\rangle$, to express what groups of agents can achieve. We have, for instance, that agents $a_1$ and $a_2$ can enforce a situation which is undesirable for $a_3$: $\mathfrak{M}, q_0 \models \langle\!\langle a_1, a_2 \rangle\!\rangle \bigcirc \mathsf{r}$. Indeed, $\{a_1, a_2\}$ and the grand coalition $\mathbb{A}\mathrm{gt}$ (since it contains $\{a_1, a_2\}$) are the only coalitions which are able to enforce $\bigcirc \mathsf{r}$; we have

$$\mathfrak{M}, q_0 \models \neg \langle\!\langle X \rangle\!\rangle \bigcirc \mathsf{r} \tag{6.1}$$

for all $X \subset \mathbb{A}\mathrm{gt}$ and $X \neq \{a_1, a_2\}$. Outcomes $\mathsf{s}$ or $\mathsf{t}$ can be enforced by $a_2$: $\mathfrak{M}, q_0 \models \langle\!\langle a_2 \rangle\!\rangle \bigcirc (\mathsf{s} \vee \mathsf{t})$. Agents $a_2$ and $a_3$ also have the ability to enforce a situation which agrees with both of their desired outcomes: $\mathfrak{M}, q_0 \models \langle\!\langle a_2, a_3 \rangle\!\rangle \bigcirc (\mathsf{s} \wedge \mathsf{t})$.

These properties do not take into account the coalitional framework, that is, whether specific coalitions can be formed or not. By using the coalitional framework, we get

$$\mathfrak{M}, q_0 \models_{\mathfrak{sem}} \langle\!\langle a_1, a_2 \rangle\!\rangle \bigcirc \mathsf{r} \wedge \neg \langle\!\![ a_1 ]\!\!\rangle \bigcirc \mathsf{r} \wedge \neg \langle\!\![ a_2 ]\!\!\rangle \bigcirc \mathsf{r}$$

for any semantics $\mathfrak{sem}$ introduced in Definition 4.44 and calculated in Example 4.52. The possible coalition (resp. coalitions) containing $a_1$ (resp. $a_2$) is $\{a_1\}$ (resp. are $\{a_2\}$ and $\{a_2, a_3\}$). But neither of these can enforce $\bigcirc \mathsf{r}$ (in $q_0$) because of (6.1). Thus, although it is the case that the coalition $\{a_1, a_2\}$ has the theoretical ability to enforce $\mathsf{r}$ in the next moment (which is a "losing" situation for $a_3$), $a_3$ should not consider it as sensible since agents $a_1$

and $a_2$ would not agree to constitute a coalition (according to the coalitional framework $\mathcal{CF}$).

The decision for a specific semantics is a crucial point and depends on the actual application. The next example shows that with respect to a particular argumentation semantics, agents are able to form a coalition which can successfully achieve a given property, whereas another argumentative semantics does not allow that.

*Example 6.55.* **CoalATL** can be used to determine whether a coalition for enforcing a specific property exists. Assume that $\mathfrak{sem}$ represents the grounded semantics. For instance, the statement

$$\mathfrak{M}, q_0 \models_{\mathfrak{sem}_{\text{complete}}} \langle\!|\emptyset|\!\rangle \bigcirc \mathsf{t}$$

expresses that there is a complete coalition (i.e. a coalition wrt to the grounded semantics) which can enforce $\bigcirc\mathsf{t}$, namely the coalition $\{a_2, a_3\}$. This result does not hold for all semantics; for instance, we have

$$\mathfrak{M}, q_0 \not\models_{\mathfrak{sem}_{\text{cs}}} \langle\!|\emptyset|\!\rangle \bigcirc \mathsf{t}$$

with respect to the coalition structure semantics, since the coalition structure is the empty coalition and $\mathfrak{M}, q_0 \not\models \langle\!\langle\emptyset\rangle\!\rangle \bigcirc \mathsf{t}$.

In the following section we sketch how the language can be extended by an *update mechanism*, in order to compare different argumentative semantics using formulae inside the object language.

### 6.5.4 An Update Mechanism

In Example 6.55 we have shown that the underlying semantics of the coalition framework is crucial for the truth of a formula. We showed, for instance, that $\langle\!|\emptyset|\!\rangle \bigcirc \mathsf{t}$ is true wrt the *complete semantics* but false regarding the *coalition structure semantics*. This comparison took place on the meta-level; two CGMs were defined, using grounded and coalition structure semantics, respectively. In this section, we introduce semantics as *first-class citizens* in the object language. Therefore, we extend the language by *semantic terms*, out of a set $\Omega$, and an update operator (**set-sem** $\cdot$). Semantically, a CGM $\mathfrak{M}$ is enriched by a *denotation function* $[\![\cdot]\!] : \Omega \to (\mathbb{CF}(\mathbb{A}\text{gt}) \to \mathcal{P}(\mathcal{P}(\mathbb{A}\text{gt})))$ which maps semantic terms to an argumentation semantics. The idea is that (**set-sem** *sem*) resets the semantics in $\mathfrak{M}$ to $[\![sem]\!]$, where $sem \in \Omega$. The intended reading for (**set-sem** *sem*)$\varphi$ is that $\varphi$ holds if the argumentation semantics is given by $[\![sem]\!]$. We formally define the new language and its models.

**Definition 6.56 ($\mathcal{L}_{CoalATL}$ plus update).** *Let $\Omega$ be a non-empty set, its elements are called semantic symbols (with typical element sem).*

*The logic $\mathcal{L}^u_{CoalATL}(\mathbb{A}\text{gt}, \Pi, \Omega)$ is given by all formulas of $\mathcal{L}_{CoalATL}(\mathbb{A}\text{gt}, \Pi)$ and for all $\varphi \in \mathcal{L}^u_{CoalATL}(\mathbb{A}\text{gt}, \Pi, \Omega)$ we also have $(\textbf{set-sem}\ sem)\varphi \in \mathcal{L}^u_{CoalATL}$, where $sem \in \Omega$.*

*Remark 6.57 (Standard semantic terms).* We assume that for all semantics defined in Definitions 4.44 and 4.47 there is a corresponding semantic term in $\Omega$. For example, for the grounded semantics $\mathfrak{sem}_{\text{grounded}}$ there is a term $sem_{\text{grounded}}$ in $\Omega$.

We need to define the denotation of the new syntactic objects.

**Definition 6.58 (CGM + update).** *A* coalitional concurrent game structure with update *(*U+CGM*) is given by a tuple*

$$\mathfrak{M} = \langle \mathbb{A}\text{gt}, Q, \Pi, \pi, Act, d, o, \zeta, \mathfrak{sem}, \Omega, [\![\cdot]\!] \rangle$$

*where $\langle \mathbb{A}\text{gt}, Q, \Pi, \pi, Act, d, o, \zeta, \mathfrak{sem} \rangle$ is a* CGM*, $\Omega$ is a non-empty set of semantic terms, and $[\![\cdot]\!] : \Omega \to (\mathbb{CF}(\mathbb{A}\text{gt}) \to \mathcal{P}(\mathcal{P}(\mathbb{A}\text{gt})))$ is a denotation function, such that $[\![sem]\!]$ is an argumentation semantics over* $\mathbb{CF}(\mathbb{A}\text{gt})$ *(cf. Definition 4.44) for all $sem \in \Omega$.*

In accordance with Remark 6.57 we assume that the denotation of semantic terms belonging to one of the "standard" semantics connects the terms with their semantics. That is, we assume, for instance, that the denotation of $sem_{\text{grounded}}$ is $\mathfrak{sem}_{\text{grounded}}$.

In addition to all semantic rules given before, we also need to interpret (**set-sem** $\cdot$).

**Definition 6.59 (Semantics).** *Let $\mathfrak{M} = \langle \mathbb{A}\text{gt}, Q, \Pi, \pi, Act, d, o, \zeta, \mathfrak{sem}, \Omega, [\![\cdot]\!] \rangle$ be a* U+CGM *and $sem \in \Omega$. The semantics of* **CoalATL** *plus update extends that of* **CoalATL**, *given in Definition 6.50 , by the following rule ($\psi \in \mathcal{L}^u_{CoalATL}(\mathbb{A}\text{gt}, \Pi, \Omega)$):*

$\mathfrak{M}, q \models (\textbf{set-sem}\ sem)\psi$ *iff* $\mathfrak{M}^{[\![sem]\!]}, q \models \psi$

*where $\mathfrak{M}^{[\![sem]\!]}$ is a* U+CGM *equal to $\mathfrak{M}$ but its argumentation semantics is given by (set to) $[\![sem]\!]$.*

*Example 6.60.* Let $\mathfrak{M}'$ be the U+CGM which corresponds to the CGM $\mathfrak{M}$ given in Example 6.55 extended by a set $\Omega$ of semantic terms and denotation function $[\![\cdot]\!]$. We can state the relation between complete and coalition structure semantics directly on the object level:

$$\mathfrak{M}, q_0 \models (\textbf{set-sem}\ \mathfrak{sem}_{\text{complete}})\langle\!|\emptyset|\!\rangle \bigcirc \mathsf{t} \wedge \neg(\textbf{set-sem}\ \mathfrak{sem}_{\text{cs}})\langle\!|\emptyset|\!\rangle \bigcirc \mathsf{t}.$$

## 6.6 Cooperation and Goals

*Why should agents join coalitions?* They must have reasons to do so. Here, we consider *goals* as the main motivation, and we assume that agents act to reach their goals. Firstly, we propose an *abstract goal framework*. Secondly, we use specific languages for goals and objectives, and we propose **ATL** as a suitable language to capture agents' goals. Finally, we *implement goals* into the semantics of **CoalATL**, discuss their benefits and illustrate it with an example.

### 6.6.1 Goals and Agents

*Pro-activeness* and *social ability* are among the widely accepted characteristics of intelligent agents [Wooldridge, 2002]. In BDI frameworks, also *goals* (or *desires*) and *beliefs* play an important role [Bratman, 1987; Rao and Georgeff, 1991].

We believe that also *the social ability to join coalitions*, should be based on some incentive. Agents are usually not developed to offer their services for free. Also in the agent programming community several types of goals (e.g. *achievement* or *maintenance* goals) are commonly considered as an agent's main motivation. Here, we present a simple abstract framework to deal with these notions.

**Definition 6.61 ($\mathcal{G}$, goal mapping $\mathfrak{g}$).** *Let $\mathcal{G}_a$ be a non-empty set of elements (set of* goals*), one for each agent $a \in \mathbb{A}\mathrm{gt}$, and $\mathcal{G} := \bigcup_{a\in\mathbb{A}\mathrm{gt}} \mathcal{G}_a$. By "g" we denote a typical element from $\mathcal{G}$. A* goal mapping *is a function $\mathfrak{g} : \mathbb{A}\mathrm{gt} \to (Q^+ \to \mathcal{P}(\mathcal{G}))$ assigning a set of goals to a given sequence of states and agent.*

So, a goal mapping assigns a set of goals to a *history*, depending on an agent. This is needed to introduce goals into CGMs. The history dependency can be used, for instance, to model when a goal should be removed from the list: An agent having a goal $\Diamond \mathsf{s}$ may drop it after reaching a state in which $\mathsf{s}$ holds. Alternatively, a model update mechanism can be used to achieve the same regarding state-based goal mappings; however, in our opinion the former seems more elegant.

An agent might have several goals. Often, goals can not be reached simultaneously which requires means to decide which goal should be selected first. We model this by a preference ordering.

**Definition 6.62 (Goal preference relation).** *A* goal preference ordering *(gp-ordering) $\preceq$ over a set of goals $\mathcal{G}' \subseteq \mathcal{G}$ is a complete, transitive, antisymmetric, and irreflexive relation $\preceq \subseteq \mathcal{G}' \times \mathcal{G}'$. We say that a goal $g$ is preferred to $g'$ if $g \preceq g'$.*

*Given a goal mapping* $\mathfrak{g}_a$ *for* $a \in \mathbb{A}\mathrm{gt}$ *we assume that there implicitly also is a gp-ordering* $\preceq_a$ *(a's gp-ordering).*

So far, we did not say how goals can be actually used to form coalitions. We assume, given some task, that agents having goals satisfied or partly satisfied by the outcome of the task are willing to cooperate to bring about the task. In the following we will use the notion *objective* (or objective formula) to refer to both the task itself and the outcome of it. A typical objective is written as $o$. Agents which have goals fulfilled or at least partly supported by objective $o$ are possible candidates to participate in a coalition aiming at $o$.

We say that an objective $o$ *satisfies* goal $g$, $o \hookrightarrow g$, if the complete goal $g$ is fulfilled after $o$ has been accomplished. If a goal is (partly) satisfied by $o$ we say that $o$ *supports* $g$, $o \hookrightarrow^s g$; i.e. there is another goal $g'$ which is a *subgoal* of $g$ and which is satisfied. These notions will be made precise in the following sections. Intuitively, an objective $\Box \mathsf{t}$ satisfies goal $\Box(\mathsf{t} \vee \mathsf{s})$.

### 6.6.2 Specifying Goals and Objectives

In this section, we propose to use $\mathcal{L}_{ATL}$-path formulas for specifying *goals*. It has been shown that temporal logics like **LTL** and **CTL** can be used as goal specification languages [Bacchus and Kabanza, 1998; Baral and Zhao, 2007; Baral et al., 2001].

Goals *formulated in* $\mathcal{L}_{LTL}$ are very intuitive. Formulae like $\Diamond \mathsf{rich}$ (*eventually being rich*), $\bigcirc \mathsf{takeUmbrella}$ (*take umbrella in the next moment*), or $\Box\Diamond \mathsf{sleep}$ (*going to sleep again and again*) have clear interpretations. But goals *formulated in* $\mathcal{L}_{CTL}$ can be ambiguous. A goal like $\mathsf{A}\Diamond\mathsf{rich}$[4] does not seem fundamentally different from $\Diamond \mathsf{rich}$ from the agent's point of view. Its goal of being rich in the future can be read implicitly as being rich in all possible futures; only one of them can actually become true and in that particular one the agent wants to be rich.

In this section we will use $\mathcal{L}_{ATL}$ for expressing agents' goals. At first glance, this seems to contradict the statement made above since $\mathcal{L}_{CTL}$ can be seen as a special case (the one agent fragment) of $\mathcal{L}_{ATL}$. But this is not the case: $\mathcal{L}_{CTL}$ refers to a purely temporal setting whereas $\mathcal{L}_{ATL}$ talks about *abilities of agents*. Here is a clarifying example. Assume that there are two agents $a$ and $b$ both having access to the same critical section; that is, either $a$ or $b$ should access this section but not both. In such a case it is reasonable that agent $a$ has the goal of preventing $b$ to enter this section on its own: $\neg\langle\langle b\rangle\rangle\Diamond\mathsf{critical}$. However, it might be acceptable for $a$ that $b$ together with another agent $c$ enters the critical section because then $c$ has to unlock resources $a$ could use instead. Let us consider a more detailed example.

[4] The operator $\mathsf{A}$ refers to all possible paths starting in a state. In **ATL** this operator can be expressed as $\langle\langle\emptyset\rangle\rangle$.

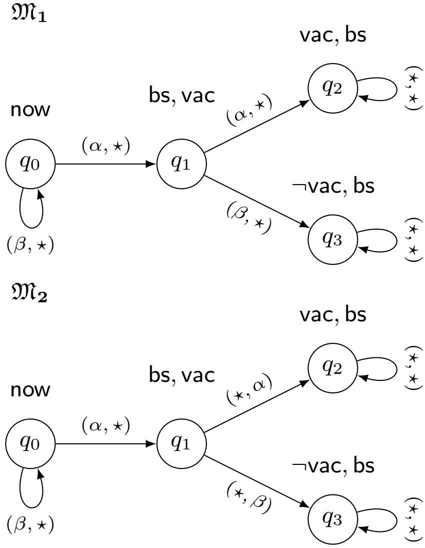



**Fig. 6.4.** Two simple models showing that **ATL** goals are useful. $\star \in \{\alpha, \beta\}$ is used as a placeholder for any of the two actions.

*Example 6.63 ($\mathcal{L}_{ATL}$-goals).* In the example we consider two agents $a$ and $b$. Both agents can perform actions $\alpha$ and $\beta$. The first agent, leader of a research group, would like to get a better salary (bs) and wants to retain the power to decide when to take vacation (vac). So, $a$'s goal can be expressed as $\gamma \equiv \Box(\neg\mathsf{now} \rightarrow \mathsf{bs} \wedge \mathsf{vac})$. Interpreting the models shown in Figure 6.4 purely temporally (i.e. without action profiles) the $\mathcal{L}_{CTL}$-formula $\mathsf{E}\gamma$ is satisfied in $q_0$ in both models: There are $q_0$-paths which satisfy $\gamma$. On the other hand, $\mathsf{A}\gamma$ is false in both models in $q_0$.

Now agent $b$ enters the stage. A higher salary would require $a$ to move to a company in which the agent has a boss who might be able to decide on $a$'s vacation (depending on the contract). Actually, although $a$ would like to have a better salary it prefers to decide on its vacation on its own. Thus, its goal can be reformulated to $\gamma' \equiv \Box(\neg\mathsf{now} \rightarrow \mathsf{bs} \wedge \neg\langle\langle b\rangle\rangle\Diamond\neg\mathsf{vac})$, or equivalently in this example $\Box(\neg\mathsf{now} \rightarrow \mathsf{bs} \wedge \langle\langle a\rangle\rangle\Box\mathsf{vac})$. Now, it is easy to see that $\mathfrak{M}_1, q_0 \models \langle\langle a\rangle\rangle\gamma'$ but $\mathfrak{M}_2, q_0 \not\models \langle\langle a\rangle\rangle\gamma'$. In the first model $b$ does not have the power to decide on $a$'s vacation but $b$ has this ability in the second model.

This quite simplistic example shows that $\mathcal{L}_{ATL}$-formulae can make sense as goal specification language.

**Definition 6.64 ($\mathcal{L}_{ATL}$-Goal).** *Let $\gamma, \gamma'$ be $\mathcal{L}_{ATL}$-path formulae. An $\mathcal{L}_{ATL}$-*goal *has the form $\gamma$ or $\gamma \wedge \gamma'$*[5]*.*

Note that goals can easily be defined as $\mathcal{L}_{CoalATL}$-formulae; however, due to simplicity we stick to pure $\mathcal{L}_{ATL}$-formulae.

It remains to define the objective language. Consider the $\mathcal{L}_{CoalATL}$-formula $\langle\!|A|\!\rangle\gamma$. The question is whether there is a *rational* group to bring about $\gamma$; thus, only agents which gain advantage when $\gamma$ is fulfilled should cooperate. Hence, we consider $\gamma$ as objective.

**Definition 6.65 ($\mathcal{L}_{CoalATL}$-objective).** *An* $\mathcal{L}_{CoalATL}$-objective *is an* $\mathcal{L}_{CoalATL}$-path *formula.*

### 6.6.3 Goals as a Means for Cooperation

In this section we combine **CoalATL** with the goal framework described above. The syntax of the logic is given as in Section 6.5.2. The necessary change takes place in the semantics. We redefine what it means for a coalition to be *valid*.

Up to now valid coalitions were solely determined by coalitional frameworks. Conflicts represented by such frameworks are a coarse, but necessary, criterion for a successful coalition formation process. However, nothing is said about incentives *to join* coalitions, only why coalitions should *not* be joined.

Goals allow to capture the first issue. For a given objective formula $o$ and a finite sequence of states, called *history*, we do only consider agents which have some goal supported by the current objective. CGMs *with goals* are given as a straightforward extension of CGMs (cf. Definition 6.47).

**Definition 6.66 (CGM with goals).** *A* CGM with goals *(*G+CGM*) $\mathfrak{M}$ is given by a model of $\mathbb{M}(Q, \mathbb{A}\mathrm{gt}, \Pi, \mathfrak{sem}, \zeta)$ extended by a set of goals $\mathcal{G}$ and a goal mapping $\mathfrak{g}$ over $\mathcal{G}$. The set of all such models is denoted $\mathbb{M}^g(Q, \mathbb{A}\mathrm{gt}, \Pi, \mathfrak{sem}, \zeta, \mathcal{G}, \mathfrak{g})$ or just $\mathbb{M}^g$ if we assume standard naming.*

To define the semantics we recall some additional notation. Given a path $\lambda \in Q^\omega$ we use $\lambda[i, j]$ to denote the sequence $\lambda[i]\lambda[i+1]\ldots\lambda[j]$ for $i, j \in \mathbb{N}_0 \cup \{\infty\}$ and $i < j$. A *history* is a finite sequence $h = q_1 \ldots q_n \in Q^+$, $h[i]$ denotes state $q_i$ if $n \geq i$, $q_n$ for $i \geq n$, and $\varepsilon$ for $i < 0$ where $i \in \mathbb{Z} \cup \{\infty\}$. Furthermore, given a history $h$ and a path or history $\lambda$ the combined path/history starting with $h$ extended by $\lambda$ is denoted by $h \circ \lambda$.

Finally, we present the semantics of **CoalATL** *with* goals. It is similar to Definition 6.50. Here, however, it is necessary to keep track of the steps (i.e. visited states) made to determine the goals of the agents which remain unsatisfied.

[5] Note that $\gamma \wedge \gamma'$ is not an $\mathcal{L}_{ATL}$-path formula anymore.

**Definition 6.67 (Goal-based semantics for $\mathcal{L}_{CoalATL}$).** *Let $\mathfrak{M}$ be a* G+CGM*, $q$ a state, $\varphi, \psi$ state-, $\gamma$ a path formula, and $i, j \in \mathbb{N}_0$. Semantics of* **CoalATL$^{+goals}$** *formulae is given as follows:*

$\mathfrak{M}, q, \tau \models \mathsf{p}$ *iff* $\mathsf{p} \in \pi(q)$,
$\mathfrak{M}, q, \tau \models \varphi \wedge \psi$ *iff* $\mathfrak{M}, q, \tau \models \varphi$ *and* $\mathfrak{M}, q, \tau \models \psi$,
$\mathfrak{M}, q, \tau \models \neg\varphi$ *iff not* $\mathfrak{M}, q, \tau \models \varphi$,
$\mathfrak{M}, q, \tau \models \langle\langle A \rangle\rangle\gamma$ *iff there is a strategy* $s_A \in \Sigma_A$ *such that for all* $\lambda \in out(q, s_A)$ *it holds that* $\mathfrak{M}, \lambda, \tau \models \varphi$,
$\mathfrak{M}, q, \tau \models \langle\!| A |\!\rangle\gamma$ *iff there is* $A' \in VC^g(q, A, \gamma, \tau)$ *such that* $\mathfrak{M}, q, \tau \models \langle\langle A \rangle\rangle\gamma$,
$\mathfrak{M}, \lambda, \tau \models \varphi$ *iff* $\mathfrak{M}, \lambda[0], \tau \models \varphi$,
$\mathfrak{M}, \lambda, \tau \models \Box\varphi$ *iff for all* $i \in \mathbb{N}_0$ *it holds that* $\mathfrak{M}, \lambda[i], \tau \circ \lambda[1, i] \models \varphi$,
$\mathfrak{M}, \lambda, \tau \models \bigcirc\varphi$ *iff it holds that* $\mathfrak{M}, \lambda[1], \tau \circ \lambda[1] \models \varphi$,
$\mathfrak{M}, \lambda, \tau \models \varphi\mathcal{U}\psi$ *iff there is a* $j \in \mathbb{N}_0$ *such that* $\mathfrak{M}, \lambda[j], \tau \circ \lambda[1, j] \models \psi$ *and for all* $0 \leq i < j$ *it holds that* $\mathfrak{M}, \lambda[i], \tau \circ \lambda[1, i] \models \varphi$.

*Ultimately, we are interested in $\mathfrak{M}, q \models \varphi$ defined as $\mathfrak{M}, q, q \models \varphi$.*

We have to define when a goal is satisfied. Although the definition of *support* can be defined similarly, we focus on the former notion only.

**Definition 6.68 (Satisfaction of goals).** *Let $g$ be an $\mathcal{L}_{ATL}$-goal, $o$ an $\mathcal{L}_{CoalATL}$-objective, and $\tau \in Q^+$. We say that objective $o$* satisfies *$g$, for short $o \hookrightarrow_{\mathfrak{M},\tau,B} g$, with respect to $\mathfrak{M}, \tau$, and $B$ if, and only if, for all strategies $s_B \in \Sigma_B$ and for all $\lambda \in out(\tau[\infty], s_B)$ it holds that $\mathfrak{M}, \lambda, \tau \models o$ implies $\mathfrak{M}, \lambda \models g$.*

A goal is satisfied by an objective if each path (enforceable by $B$) that satisfies the objective does also satisfy the goal. That is, satisfaction of the objective will guarantee that the goal becomes true.

All the new functionality provided by goals is captured by the new valid coalition function $\mathrm{VC}^g$

**Definition 6.69 (Valid coalitions, $\mathbf{VC}^g(q, A, o, \tau)$).** *Let $\mathfrak{M} \in \mathbb{M}^g$, $\tau \in Q^+$, $A, B \subseteq \mathbb{A}\mathrm{gt}$, $o$ an $\mathcal{L}_{CoalATL}$-objective.*

*We say that $B$ is a* valid coalition *after $\tau$ with respect to $A$, $o$, and $\mathfrak{M}$ if, and only if,*

1. *$B \in \mathfrak{sem}(\zeta(A)(\tau[\infty]))$, $A \cap B \neq \emptyset$ if $A \neq \emptyset$, and*
2. *there are goals $g_{b_i} \in \mathfrak{g}_{b_i}(\tau)$, one per agent $b_i \in B$, such that $o \hookrightarrow_{\mathfrak{M},\tau,B} g_{b_1} \wedge \cdots \wedge g_{b_{|B|}}$*

*The set $VC^g(q, A, o, \tau)$ consists of all such valid coalitions wrt to $\mathfrak{M}$.*

Thus, for the definition of valid coalitions among other things, a goal mapping, a function $\zeta$ and a sequence of states $\tau$ are required. The intuition of $\tau$

is that it represents the history (the sequence of states visited so far including the current state). So, $\tau$ is used to determine which goals of the agents are still active.

**Proposition 6.70.** *If* $\mathfrak{M}, q, \tau \models \langle\!| A |\!\rangle \gamma$ *then there is a coalition* $B \in VC(A, q)$ *and goals* $g_b \in \mathfrak{g}_b(\tau)$ *one for each* $b \in B$ *such that* $\mathfrak{M}, q, \tau \models \langle\!\langle B \rangle\!\rangle (\gamma \wedge \bigwedge_{b \in B} g_b)$.

*Proof.* Suppose $\mathfrak{M}, q, \tau \models \langle\!| A |\!\rangle \gamma$. Then, there is a coalition $B \in \mathrm{VC}^g(q, A, o, \tau)$ such that $\mathfrak{M}, q, \tau \models \langle\!\langle B \rangle\!\rangle \gamma$ iff $B \in \mathfrak{sem}(\zeta(A)(\tau[\infty]))$, $A \cap B \neq \emptyset$ if $A \neq \emptyset$ and there are goals $g_{b_i} \in \mathfrak{g}_{b_i}(\tau)$, one per agent $b_i \in B$, such that $\gamma \hookrightarrow_{\mathfrak{M}, \tau, B} g_{b_1} \wedge \cdots \wedge g_{b_{|B|}}$ iff there is a $B \in \mathfrak{sem}(\zeta(A)(\tau[\infty]))$, $A \cap B \neq \emptyset$ if $A \neq \emptyset$ and a strategy $s_B$ such that for all $\lambda \in out(q, s_B)$, $\mathfrak{M}, \lambda, \tau \models \gamma$ and there are goals $g_{b_i} \in \mathfrak{g}_{b_i}(\tau)$, one per agent $b_i \in B$, such that for all strategies $s'_B \in \Sigma_B$ and all $\lambda \in out(\tau[\infty], s'_B)$ it holds that $\mathfrak{M}, \lambda, \tau \models \gamma$ implies $\mathfrak{M}, \lambda \models g$. We have that $\gamma$ is satisfied on all paths resulting from $s_A$; hence, also $g_{b_1} \wedge \cdots \wedge g_{b_{|B|}}$ and we have that there is a $B \in \mathfrak{sem}(\zeta(A)(\tau[\infty]))$, $A \cap B \neq \emptyset$ if $A \neq \emptyset$ and a strategy $s_B$ and goals $g_{b_i} \in \mathfrak{g}_{b_i}(\tau)$, one per agent $b_i \in B$, such that for all $\lambda \in out(q, s_B)$, $\mathfrak{M}, \lambda, \tau \models \gamma \wedge \bigwedge_{b \in B} g_b$. ■

*Remark 6.71 (Preferred goals).* In the abstract goal framework presented in Section 6.6.1 we defined a preference ordering over goals. The gp-orderings highly influence the coalition formation process. However, for this paper we decided to focus on the pure goal framework since the interplay between the formation process becomes much more sophisticated if preferences are taken into account. We just give a brief motivation for preferences and why they increase the complexity of coalition building.

The set of valid coalitions consists of all coalitions which are acceptable/conflict-free (according to a coalitional framework) and in which all agents have an incentive to join the coalition (that is, some goal has to be satisfied/supported). Let us consider two valid groups $B$ and $B'$ both containing the agent $a$. Both groups are somewhat appealing for $a$ since they satisfy some of his goals, say $B$ (resp. $B'$) can bring about $g$ (resp. $g'$). In our framework $B$ and $B'$ are treated equally good. Is this reasonable? From an abstract level it is; however, a finer grained analysis should incorporate the preferences between goals. If, for instance, $g$ is preferred over $g'$ agent $a$ should rather go for coalition $B$ instead of $B'$. The agent would prefer to bring about $g$ thus joining $B$. On the other hand, if $a$ refuses to join $B'$ it might be possible, by a symmetric argument, that another agent, say $b$, refuses to take part in $B$, such that in the end neither $B$ nor $B'$ will form. Of course, in such a situation both agents prefer to bring about their less preferred goals. This is still better than getting nothing.

This reasoning very much reminds on game theoretic rationality concepts. For example, the motivation behind a Nash equilibrium strategy shows a strong connection: No agent has an incentive to unilaterally choose another strategy. Even closer are concepts from cooperative game theory. This discussion shows that the incorporation of a preference ordering over goals is quite interesting.

### 6.6.4 Progression of $\mathcal{L}_{ATL}$-goals

A goal mapping takes the history into account to be able to reflect if a goal has become fulfilled. For example, if an agent has goal $\Diamond \mathsf{p}$ and $\mathsf{p}$ became satisfied in a state on the current history the goal should be marked as completed in the following state. (Of course, a new goal in this state can again be $\Diamond \mathsf{p}$.) Another, more practical but also restricted option, is to consider an initial goal base $\mathcal{GB}$ and modify, specialise or remove, the formulae according to the steps taken. So, goal $\Diamond \mathsf{p} \wedge \Box \mathsf{q}$ should be specialised to $\Box \mathsf{q}$ if a state is reached in which $\mathsf{p}$ holds. In [Bacchus and Kabanza, 2000] such a progression procedure is presented for first-order linear time temporal logic.

## 6.7 Summary

We have proposed logics to reason about abilities of *rational agents* under perfect information, among them **ATLP** and **CoalATL**. The first logic can be used to study the outcome of rational play in a logical framework, under various rationality criteria. To our knowledge, there has been very little work on this issue (although solving game-like scenarios with help of various solution concepts is arguably the main application of game theory). We note that we are *not* discussing the merits of this or that rationality criterion here, nor the pragmatics of using particular criteria to predict the actual behaviour of agents. Our aim was to propose a conceptual tool to study the consequences of accepting one or another criterion.

We believe that the logic we have proposed provides much flexibility and modelling power. The results presented in Sections 6.4 and 10.1 also suggest that the expressiveness of the language is quite high. We have discussed how **ATLP** can be used to describe the two kinds of solution concepts from Section 3.3 in a uniform way.

We have constructively shown that several logics can be embedded into **ATLP**. That is, we have demonstrated how models and formulae of those logics can be (independently) transformed to their **ATLP** counterparts in a way that preserves truth.

We have extended the results from [van der Hoek et al., 2005a; Jamroga et al., 2005] presented in Section 3.3, and have shown that the classical solution concepts (*Nash equilibrium*, *subgame perfect Nash equilibrium*, *Pareto optimality*, and others) can be also characterised in $\mathcal{L}_{ATLP}$ in a uniform way.

We have proposed expressions in $\mathcal{L}_{ATLP}$ that, given an extensive form game, denote exactly the set of Nash equilibria (subgame perfect NE's, Pareto optimal profiles, etc.) in that game. As a consequence, **ATLP** can serve both as a language for reasoning about rational play, and for specifying what rational play is. We have pointed out that these characterisations extend traditional solution concepts to the more general class of multi-stage multi-player games defined by concurrent game structures. Similarly, we have considered general solution concepts from Section 3.3.2

In Chapter 7 we present an extension of **ATLP** which allows us to study strategies, time, knowledge, and plausible/rational behaviour under both perfect and *imperfect information*. It turns out that putting so many dimensions in one framework has many side effects – even more so in this case because the interaction between abilities and knowledge is non-trivial (cf. [Jamroga and van der Hoek, 2004; Jamroga and Ågotnes, 2006; Herzig and Troquard, 2006]). In [Bulling and Jamroga, 2007a], we have investigated *time, knowledge and plausibility*.

The second logic that has been presented is **CoalATL** an extension of **ATL** which is able to model coalition formation through argumentation. Our formalism includes two different modalities, $\langle\langle A\rangle\rangle$ and $\langle\!| A |\!\rangle$, which refer to different kinds of abilities agents may have. Note that the original operator $\langle\langle A\rangle\rangle$ is used to reason about the pure ability of the very group $A$. However, the question whether it is reasonable to assume that the members of $A$ collaborate is not taken into account in **ATL**. With the new operator $\langle\!| A |\!\rangle$ we have tried to close this gap, providing also a way to focus on *sensible coalition structures*. In this context, "sensible" refers to *acceptable coalitions with respect to some argumentative semantics* (as characterised in Def. 4.44).

Furthermore, we have defined the formal machinery required for characterising argument-based coalition formation in terms of the proposed operator $\langle\!| A |\!\rangle$. Coalitions can be actually computed in terms of a given argumentation semantics, which can be given as a parameter within our model, thus providing a modular way of analysing the results associated with different alternative semantics. This has allowed us to compare the ability of agents to form particular coalitions and study emerging properties regarding different semantics. In Section 10.2 the model checking algorithm used in **ATL** is extended to model check **CoalATL**.

# 7

# Rational Agents: Imperfect Information

## Contents



In Section 6 we have considered how the behaviour of rational agents can be modelled and analysed. The setting was restricted to agents being aware of the current state of the world. In this section we turn to *incomplete information* settings. We consider two different approaches to incomplete information.

In the first setting, we present *constructive strategic logic with plausibility* (**CSLP**), a combination of **CSL** (compare Section 2.3.2) and **ATLP** (Section 6.2) where the new language goes far beyond the pure union of both logics. The plausibility concept allows us to neatly define the relationship between epistemic and doxastic concepts. It allows to analyse rational play under incomplete information where the latter notion refers to the indistinguishability of states.

In the second proposal, we focus on rational play of a group of agents which has some (incomplete) prediction about the opponents' behaviour. More precisely, usually it is assumed that the opponents behave in the most destructive way. However, there are scenarios were such an assumption is not reasonable; for example, due to the lack of communication channels. We model this by assuming some *probabilistic behaviour* of the opponents. The proposed logic introduces a notion of "randomness" to the responses of the opponents; therefore, we named the logic *alternating time temporal logics with probabilistic success.*

## 7.1 Rational Play under Incomplete Information: CSLP

The logic **CSL** presented in Section 2.3.2 unified several attempts to incorporate epistemic concepts into **ATL**, and solved problems of these previous attempts. However, it includes only strategic and epistemic modalities; in particular, doxastic and rationality concepts are absent. On the other hand, **ATLP** introduced in Chapter 6 allows for reasoning about *rational* or *plausible* behaviour. In this chapter we combined both ideas. The resulting logic *constructive strategic logic with plausibility* (**CSLP**) allows to reason about strategic ability and rational behaviour under incomplete information. We use the plausibility concept to define the relationship between epistemic and doxastic concepts, in a similar way to the logic **CTLKP** from [Bulling and Jamroga, 2007a]. This logic is a result of extending **CTLK** [Penczek and Lomuscio, 2003], a direct combination of the branching time logic **CTL** [Emerson, 1990] and standard epistemic logic [Fagin et al., 1995], by a notion of plausibility which in turn was used to define a particular notion of beliefs. Plausibility assumptions were defined in terms of paths in the underlying system. Then, an agent's beliefs were given by its knowledge if only plausible paths were considered. The idea to build beliefs on top of plausibility has been inspired by [Su et al., 2005; Friedman and Halpern, 1994]. Another source of inspiration is [van der Hoek et al., 2004; van Otterloo and Jonker, 2004], where the semantics of ability was influenced by particular notions of rationality.

As the basic modalities of **CSLP** we introduce *weak constructive rational beliefs*: $\mathbb{CW}_A$ (common beliefs), $\mathbb{DW}_A$ (distributed beliefs), and $\mathbb{EW}_A$ (mutual beliefs). The term *constructive* is used in the same sense as in **CSL**, where it referred to an "operational" kind of knowledge that, in order to "know how to play", requires the agents to be able to *identify* and *execute* an appropriate strategy. Like for **CSL**, the semantics of **CSLP** is non-standard: Formulae are interpreted in *sets of states*. For example, the intuitive reading of $\mathfrak{M}, Q' \models \langle\langle A \rangle\rangle \gamma$ is that agents $A$ have a collective strategy which enforces $\gamma$ from *each* state in $Q'$. Thanks to the plausibility concept provided by **ATLP** we can define *knowledge* and *rational beliefs* on top of weak beliefs. We point out that our notion of rational belief is rather specific, and show interesting properties of knowledge, rational belief, and plausibility. In particular, it is shown that knowledge and belief are **KD45** modalities.

We show that **CSLP** is very expressive, and we demonstrate how solution concepts for imperfect information games can be characterised and used in **CSLP**. It also turns out that, despite the logic's expressiveness, the model checking complexity does not increase when compared to a specific fragment of **ATLP**, and increases only slightly compared to **CSL** when plausibility and rational beliefs are added (cf. Section 10.3) .

In summary, **CSLP** is an attempt to integrate the notions of time, knowledge, belief, strategic ability, rationality, and uncertainty in a single logical framework.

### 7.1.1 Agents, Beliefs, and Rational Play

In this section we informally describe and summarise the ingredients of **CSLP**. In the following, let $A \subseteq \mathbb{A}\text{gt}$ be a team of agents. Formulae of **CSLP** are interpreted given a model $\mathfrak{M}$ and a set of states $Q'$ (as in the case of **CSL**, cf Section 2.3.2). The reading of $\mathfrak{M}, Q' \models \langle\langle A \rangle\rangle \gamma$ is that agents $A$ have a collective strategy which enforces $\gamma$ from *all* states in $Q'$. $\mathbf{Pl}_A\, \varphi$ assumes that agents in $A$ play plausibly according to some rationality criterion which can be set (resp. refined) by operators (**set-pl** $\omega$) (resp. (**refn-pl** $\omega$)). The set of such *rational agents* is denoted by $\mathbb{R}\text{gt}$. Plausibility terms $\omega$ refer to sets of strategy profiles that implement the rationality criteria. Finally, the logic includes operators for *constructive weakly rational belief* (*constructive weak belief*/CWB in short):

- $\mathbb{CW}_A\varphi$ (agents $A$ have common CWB in $\varphi$);
- $\mathbb{EW}_A\varphi$ (agents $A$ have mutual CWB in $\varphi$); and
- $\mathbb{DW}_A\varphi$ (agents $A$ have distributed CWB in $\varphi$).

Semantically, the CWB operators yield "epistemic positions" of team $A$ that serve as reference for the semantic evaluation of strategic formulae.

Let us consider $\mathfrak{M}, Q' \models \mathbb{EW}_A \mathbf{Pl}_{\mathbb{Agt}\setminus A} \langle\langle A \rangle\rangle \Box \mathsf{safe}$ (*coalition A has constructive mutual weak belief that they can keep the system safe forever if the opponents behave rationally*) in model $\mathfrak{M}$ and set of states $Q'$. Firstly, $Q'$ is extended with all states indistinguishable from some state in $Q'$ for any agent from $A$. Let us call the extended set $Q''$. Agents in $A$ have CWB in $\mathbf{Pl}_{\mathbb{Agt}\setminus A} \langle\langle A \rangle\rangle \Box \mathsf{safe}$ iff they have a strategy that maintains $\mathsf{safe}$ from all states in $Q''$ assuming that implausible behaviour of agents in $\mathbb{Agt} \setminus A$ is disregarded.

Later, we will define strongly rational beliefs (resp. knowledge) as a special case of CWBs in which all agents are (resp. no agent is) assumed to play plausibly.

### 7.1.2 The Language $\mathcal{L}_{CSLP}^{base}$

We proceed similar to Sections 6.1 and 6.2 and define a hierarchy of languages/logics. We begin with the base language $\mathcal{L}_{CSLP}^{base}$. It includes atomic propositions, Boolean connectives, strategic formulae, operators for *constructive weakly rational beliefs*, and operators that handle *plausibility updates*. As we will see, standard/constructive strongly rational beliefs and knowledge can be defined on top of these.

**Definition 7.1 ($\mathcal{L}_{CSLP}^{base}$).** *Let $\Omega$ be a set of primitive plausibility terms. The logic $\mathcal{L}_{CSLP}^{base}(\mathbb{Agt}, \Pi, \Omega)$ is generated by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle \bigcirc \varphi \mid \langle\langle A \rangle\rangle \Box \varphi \mid \langle\langle A \rangle\rangle \varphi \mathcal{U} \varphi \mid \mathbb{CW}_A \varphi \mid \mathbb{EW}_A \varphi \mid \mathbb{DW}_A \varphi \mid \mathbf{Pl}_A \varphi \mid (\textbf{\textit{set-pl}}\ \omega)\varphi \mid (\textbf{\textit{refn-pl}}\ \omega)\varphi.$$

The temporal operators $\bigcirc, \Box$, and $\mathcal{U}$ have their usual meaning. Moreover we define

- $\mathsf{Now}\varphi \equiv \varphi \mathcal{U} \varphi$ (now),
- $\mathbf{Pl} \equiv \mathbf{Pl}_{\mathbb{Agt}}$ (reasoning under the assumption that all agents behave plausibly), and
- $\mathbf{Ph} \equiv \mathbf{Pl}_{\emptyset}$ (reasoning about outcome of all "physically" possible behaviours).

Constructive weak belief operators for individual agents and standard weak belief operators are defined as follows:

- $\mathsf{Now}\varphi \equiv \varphi \mathcal{U} \varphi$ (now),
- $\mathbb{W}_a \varphi \equiv \mathbb{CW}_{\{a\}} \varphi$ (individual CWB),
- $\mathsf{CW}_A \varphi \equiv \mathbb{CW}_A \langle\langle \emptyset \rangle\rangle \mathsf{Now}\varphi$, $\mathsf{EW}_A \varphi \equiv \mathbb{EW}_A \langle\langle \emptyset \rangle\rangle \mathsf{Now}\varphi$, $\mathsf{DW}_A \varphi \equiv \mathbb{DW}_A \langle\langle \emptyset \rangle\rangle \mathsf{Now}\varphi$ (standard weak belief, WB),
- $\mathsf{W}_a \varphi \equiv \mathsf{CW}_{\{a\}} \varphi$ (individual WB).

Finally, we define operators for constructive and standard *strongly rational belief* (CRB) as:

$$\mathbb{B}\mathsf{el}_a \equiv \mathbb{W}_a\mathbf{Pl}, \qquad \mathbb{C}\mathbb{B}\mathsf{el}_A \equiv \mathbb{C}\mathbb{W}_A\mathbf{Pl},$$
$$\mathbb{E}\mathbb{B}\mathsf{el}_A \equiv \mathbb{E}\mathbb{W}_A\mathbf{Pl}, \qquad \mathbb{D}\mathbb{B}\mathsf{el}_A \equiv \mathbb{D}\mathbb{W}_A\mathbf{Pl},$$
$$\mathsf{Bel}_a \equiv \mathbf{Ph}\,\mathsf{W}_a\mathbf{Pl}, \qquad \mathsf{CBel}_A \equiv \mathbf{Ph}\,\mathsf{CW}_A\mathbf{Pl},$$
$$\mathsf{EBel}_A \equiv \mathbf{Ph}\,\mathsf{EW}_A\mathbf{Pl}, \; \mathsf{DBel}_A \equiv \mathbf{Ph}\,\mathsf{DW}_A\mathbf{Pl},$$

and the constructive and standard *knowledge* operators as:

$$\mathbb{K}_a \equiv \mathbf{Ph}\,\mathbb{W}_a, \quad \mathbb{C}_A \equiv \mathbf{Ph}\,\mathbb{C}\mathbb{W}_A, \; \mathbb{E}_A \equiv \mathbf{Ph}\,\mathbb{E}\mathbb{W}_A,$$
$$\mathbb{D}_A \equiv \mathbf{Ph}\,\mathbb{D}\mathbb{W}_A, \; \mathsf{K}_a \equiv \mathbf{Ph}\,\mathsf{W}_a, \quad \mathsf{C}_A \equiv \mathbf{Ph}\,\mathsf{CW}_A,$$
$$\mathsf{E}_A \equiv \mathbf{Ph}\,\mathsf{EW}_A, \; \mathsf{D}_A \equiv \mathbf{Ph}\,\mathsf{DW}_A.$$

In Section 7.1.4 we show that these definitions capture the respective notions of knowledge and belief appropriately.

### 7.1.3 Semantics: The Logic CSLP$^{\mathsf{base}}$

ICGSs from Definition 2.27 extended with plausibility (cf. Definition 6.2) serve as models for $\mathcal{L}^{base}_{CSLP}$. The relations $\sim^E_A$, $\sim^C_A$ and $\sim^D_A$, used to model group epistemics, are derived from the individual relations of agents from $A$ as defined in Definition 2.40.

Here, we use the notion *strategy* $s_a$ of agent $a$ to refer to an *ir*-strategy from Definition 2.29 (i.e. to a memoryless imperfect information or to a uniform strategy). That is, $q \sim_a q'$ implies $s_a(q) = s_a(q')$. We also use the other notations introduced so far.

**Definition 7.2** (ICGSP, **plausibility model).** *An* imperfect information concurrent game structure with plausibility *(*ICGSP*) is given by*

$$\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \sim_1, \ldots, \sim_k, \Upsilon, \mathbb{R}\mathrm{gt}, \Omega, [\![\cdot]\!]\rangle,$$

*where*

- $\langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \sim_1, \ldots, \sim_k\rangle$ *is an* ICGS *(cf. Definition 2.27),*
- $\Upsilon \subseteq \Sigma$ *is a set of plausible strategy profiles (called* plausibility set*),*
- $\mathbb{R}\mathrm{gt} \subseteq \mathbb{A}\mathrm{gt}$ *is a set of* rational agents *(i.e., the agents to whom the plausibility assumption will apply),*
- $\Omega$ *is a set of plausibility terms, and*
- $[\![\cdot]\!] : \mathcal{P}(Q) \to (\Omega \to \Sigma)$ *is a* plausibility mapping *that provides the denotation of the terms.*[1]

*We refer to* $(\Upsilon, \mathbb{R}\mathrm{gt})$ *as the* plausibility model *of* $\mathfrak{M}$*. When necessary, we write* $X_{\mathfrak{M}}$ *to denote the element* $X$ *of model* $\mathfrak{M}$*.*

---

[1] In this section, the denotation of such terms is fixed; in Section 7.1.5 we present a more flexible version.

Note, that differently from Section 6.1.4 the denotation depends on sets of states rather than single states.

*Remark 7.3 (Rational group of agents).* Differently to **ATLP** we have added the group of rational agents $\mathbb{R}\text{gt}$ to the model rather than annotating the satisfaction relation. Both approaches are equivalent but we believe that adding the agents to the models turns the focus to the new epistemic and doxastic concepts.

Imposing strategic restrictions on a *subset* $\mathbb{R}\text{gt}$ of agents can be desirable due to several reasons. It might, for example, be the case that only information about the proponents' play is available; hence, assuming plausible behaviour of the opponents is neither sensible nor justified. Or, even simpler, a group of (simple minded) agents might be known not to behave rationally.

Consider now formula $\langle\langle A\rangle\rangle\gamma$: Team $A$ tries to execute a strategy that brings about $\gamma$, but the members of the team who are also in $\mathbb{R}\text{gt}$ can only choose plausible strategies. The same applies to $A$'s opponents that are contained in $\mathbb{R}\text{gt}$. So it is exactly as for **ATLP**. Due to the plausibility model we can simplify the notation of $B$-plausible strategies and their outcome (cf. Definition 6.5 and 6.7).

**Definition 7.4 (Plausibility of strategies).** *We say that $s_A$ is* plausible *iff it is $\mathbb{R}\text{gt}$-plausible (in the sense of Definition 6.5).*

*By $\Sigma^*$ we denote the set of all plausible strategy profiles in which $\mathbb{R}\text{gt}$'s substrategy is plausible; i.e. $\Sigma^* = \Upsilon_{\mathfrak{M}}(\mathbb{R}\text{gt})$.*

*Remark 7.5.* Analogously to Remark 6.6 and from Definition 6.4 we have that every profile is $\emptyset$-plausible; i.e. $\Sigma^* = \Upsilon_{\mathfrak{M}}(\emptyset)$ for $\mathbb{R}\text{gt} = \emptyset$. We also have that $\Sigma^* = \Upsilon_{\mathfrak{M}}(\mathbb{R}\text{gt}) = \emptyset$ for $\mathbb{R}\text{gt} \neq \emptyset$ and $\Upsilon = \emptyset$.

Similarly, we simplify the notion of the outcome.

**Definition 7.6 (Plausible outcome paths).** *The* plausible outcome*,*

$$out(q, s_A, \Sigma^*),$$

*of strategy $s_A$ from state $q$ is defined as the set of paths (starting from $q$) which can occur when only plausible strategy profiles can be played and agents in $A$ follow $s_A$; that is, $out(q, s_A, \Sigma^*) = \{\lambda \in \Lambda(q) \mid \exists t \in \Sigma^* \ (t|_A = s_A \text{ and } out(q,t) = \{\lambda\})\}$.*

*In the following we will just write $out(q, s_A)$ to refer to the plausible outcome $out(q, s_A, \Sigma^*)$ if clear from context.*

*Remark 7.7.* Note, that for $\mathbb{R}\text{gt} = \emptyset$ we have that the plausible outcome is equal to the standard outcome of ATLs; i.e., $out(q, s_A, \Sigma^*) = out(q, s_A)$. Moreover, for $\mathbb{R}\text{gt} \neq \emptyset$ and $\Upsilon = \emptyset$ we have that $out(q, s_A, \emptyset) = \emptyset$.

We define the notion of *formula $\varphi$ being satisfied* by a (non-empty) set of states $Q'$ in model $\mathfrak{M}$, written $\mathfrak{M}, Q' \models \varphi$. We will also write $\mathfrak{M}, q \models \varphi$ as a shorthand for $\mathfrak{M}, \{q\} \models \varphi$. It is the latter notion of satisfaction (in single states) that we are ultimately interested in–but it is defined in terms of the (more general) satisfaction in sets of states. As in Section 2.3.2 let $\mathrm{img}(q, \mathcal{R})$ be the image of state $q$ with respect to binary relation $\mathcal{R}$, i.e., the set of all states $q'$ such that $q\mathcal{R}q'$. Moreover, we use $out(Q', s_A)$ as a shorthand for $\bigcup_{q \in Q'} out(q, s_A)$, and $\mathrm{img}(Q', \mathcal{R})$ as a shorthand for $\bigcup_{q \in Q'} \mathrm{img}(q, \mathcal{R})$. The semantics of **CSLP** is a simple combination of the one for **CSL** (cf. Definition 2.40) and **ATLP** (cf. Definition 6.8).

**Definition 7.8 (Semantics, CSLP$^{\text{base}}$).** *Let $\mathfrak{M}$ be an* ICGSP*, $Q' \subseteq Q_{\mathfrak{M}}$ be a set of states. The semantics for $\mathcal{L}^{base}_{CSLP}$ is defined as follows:*

$\mathfrak{M}, Q' \models \mathsf{p}$ *iff* $\mathsf{p} \in \pi(q)$ *for every* $q \in Q'$;
$\mathfrak{M}, Q' \models \neg\varphi$ *iff* $\mathfrak{M}, Q' \not\models \varphi$;
$\mathfrak{M}, Q' \models \varphi \land \psi$ *iff* $\mathfrak{M}, Q' \models \varphi$ *and* $\mathfrak{M}, Q' \models \psi$;
$\mathfrak{M}, Q' \models \langle\langle A\rangle\rangle \bigcirc \varphi$ *iff there exists* $s_A \in \Sigma^*|_A$ *such that, for each* $\lambda \in out(Q', s_A)$, *we have that* $\mathfrak{M}, \{\lambda[1]\} \models \varphi$;
$\mathfrak{M}, Q' \models \langle\langle A\rangle\rangle \Box \varphi$ *iff there exists* $s_A \in \Sigma^*|_A$ *such that, for each* $\lambda \in out(Q', s_A)$ *and* $i \geq 0$, *we have* $\mathfrak{M}, \{\lambda[i]\} \models \varphi$;
$\mathfrak{M}, Q' \models \langle\langle A\rangle\rangle \varphi \mathcal{U} \psi$ *iff there exists* $s_A \in \Sigma^*|_A$ *such that, for each* $\lambda \in out(Q, s_A)$, *there is an* $i \geq 0$ *for which* $\mathfrak{M}, \{\lambda[i]\} \models \psi$ *and* $\mathfrak{M}, \{\lambda[j]\} \models \varphi$ *for every* $0 \leq j < i$.
$\mathfrak{M}, Q' \models \hat{\mathcal{K}}\mathbb{W}_A \varphi$ *iff* $\mathfrak{M}, \mathrm{img}(Q', \sim^{\mathcal{K}}_A) \models \varphi$ *(where* $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ *and* $\mathcal{K} = C, E, D$*, respectively).*
$\mathfrak{M}, Q' \models \mathbf{Pl}_A\, \varphi$ *iff* $\mathfrak{M}', Q' \models \varphi$, *where the new model* $\mathfrak{M}'$ *is equal to* $\mathfrak{M}$ *but the new set* $\mathbb{R}\mathrm{gt}_{\mathfrak{M}'}$ *of rational agents in* $\mathfrak{M}'$ *is set to* $A$.
$\mathfrak{M}, Q' \models (\textbf{\textit{set-pl}}\ \omega)\varphi$ *iff* $\mathfrak{M}', Q' \models \varphi$ *where* $\mathfrak{M}'$ *is equal to* $\mathfrak{M}$ *with* $\Upsilon_{\mathfrak{M}'}$ *set to* $[\![\omega]\!]^{Q'}_{\mathfrak{M}}$.
$\mathfrak{M}, Q' \models (\textbf{\textit{refn-pl}}\ \omega)\varphi$ *iff* $\mathfrak{M}', Q' \models \varphi$ *where* $\mathfrak{M}'$ *is equal to* $\mathfrak{M}$ *with* $\Upsilon_{\mathfrak{M}'}$ *set to* $\Upsilon_{\mathfrak{M}} \cap [\![\omega]\!]^{Q'}_{\mathfrak{M}}$.

As in **CSL**, we use two notions of validity, *weak* and *strong*, depending on whether formulae are evaluated with respect to single states or sets of states.

**Definition 7.9 (Validity).** *We say that $\varphi$ is* valid *if $\mathfrak{M}, q \models \varphi$ for all* ICGSP *$\mathfrak{M}$ with plausibility model $(\Sigma, \emptyset)$ (i.e. all strategies are assumed to be plausible and no agent plays plausibly yet) and all states $q \in Q_{\mathfrak{M}}$.*

*In addition to that, we say that $\varphi$ is* strongly valid *if $\mathfrak{M}, Q' \models \varphi$ for* all ICGSP *$\mathfrak{M}$ and all sets of states $Q' \subseteq Q_{\mathfrak{M}}$.*

Note that strong validity is interpreted in *all* models and not only in those with plausibility model $(\Sigma, \emptyset)$. This stronger notion is necessary for interchangeability of (sub)formulae. The following results are straightforward.

**Proposition 7.10.** *Strong validity implies validity.*

**Proposition 7.11.** *If $\varphi_1 \leftrightarrow \varphi_2$ is strongly valid, and $\psi'$ is obtained from $\psi$ through replacing an occurrence of $\varphi_1$ by $\varphi_2$, then $\mathfrak{M}, Q' \models \psi$ iff $\mathfrak{M}, Q' \models \psi'$.*

*Proof.* The proof is done in a straightforward way by structural induction on $\psi$. ■

We also say that $\varphi$ is *satisfiable* if $\mathfrak{M}, q \models \varphi$ for some ICGSP with plausibility model $(\Sigma, \emptyset)$.

### 7.1.4 Knowledge and Rational Beliefs

In this section we motivate the logic's epistemic and doxastic operators. We show that the syntactic definitions for the derived knowledge and belief operators have an intuitive semantics.

#### Knowledge

The concept of knowledge is very simple: It is about everything which is "physically" possible, i.e., *all* behaviours are taken into account (not only the plausible ones). In particular this means that, once a knowledge operator occurs, the set of rational agents in the plausibility model becomes void, indicating that *no* agent is assumed to play rationally.

#### Weakly and Strongly Rational Beliefs

Constructive weak beliefs (CWB) ("common belief", "distributed belief", and "mutual belief") are primitive operators in our logic. All other belief/knowledge operators are derived from CWB and plausibility. In this section, we mainly discuss individual knowledge and beliefs, but the analysis extends to collective attitudes in a straightforward way.

Let us for example consider the individual CWB operator $\mathbb{W}_a\varphi$, with the following reading: Agent $a$ has *constructive weak belief* in $\varphi$ iff $\varphi$ holds in all states that $a$ considers possible, where all agents behave according to the currently specified plausibility model $(\Upsilon, A)$. That is, agents in $A$ are assumed to play as specified in $\Upsilon$. It is important to note that *weakly rational* beliefs restrict *only* the behaviour of the agents specified in the current plausibility model (i.e. $A$). This is the difference between weak and strong belief – the latter assume plausible behaviour of *all* agents. This is why we call such beliefs *strongly rational*, as it restricts the behaviour of the system in a more rigorous way due to stronger rationality assumptions.

Using rationality assumptions to define beliefs makes them quite specific. They differ from most "standard" concepts of belief in two main respects. Firstly, our notion of beliefs is focused on *behaviour* and *abilities* of agents. When no action is considered, all epistemic and doxastic notions coincide.

**Proposition 7.12.** *Let $\varphi$ be a propositional formula. Then, $\mathbb{W}_a\varphi \leftrightarrow \mathbb{B}\mathsf{el}_a\varphi \leftrightarrow \mathbb{K}_a\varphi$ is strongly valid.*

*Proof.* The notions of beliefs and knowledge only differ in the way they address the set of plausible strategies. However, plausible strategies do only affect strategic properties. Since $\varphi$ is propositional and the set of considered states does not change, its truth value is the same for the three operators. ■

Secondly, rational beliefs are about *restricting the expected behaviour* due to rationality assumptions: Irrational behaviours are simply disregarded. To strengthen this important point consider the following statements:

(i) *Ann (a) knows that Bill (b) can commit suicide* (which can be formalised as $\mathbb{K}_a\langle\langle b\rangle\rangle\Diamond\mathsf{suicide}$);
(ii) *Ann constructively believes that Bill can commit suicide* (which we tentatively formalise as $\mathbb{B}\mathsf{el}_a\langle\langle b\rangle\rangle\Diamond\mathsf{suicide}$).

In the usual treatment of beliefs, statement (i) does not imply statement (ii), but this does not hold for *rational* beliefs. That is because, typically, beliefs and knowledge are both about "hard facts". Thus, if $a$ knows some fact to be true, it should also include it in its belief base. On the other hand, our reading of $\mathbb{B}\mathsf{el}_a\langle\langle b\rangle\rangle\Diamond\mathsf{suicide}$ is given as follows: If all agents are constrained to act rationally then Ann knows a strategy for Bill by which he can commit suicide. However, it is natural to assume that no rational entity would commit suicide.[2] Hence, Bill's ability to commit suicide is out of question if we assume him to act rationally. Such an irrational behaviour is just unthinkable and thus disregarded by Ann! While she knows that Bob can commit suicide in general, she has no *plausible* recipe for Bob to do that.

A similar analysis can be conducted for standard (i.e., non-constructive) beliefs. Consider the following variants of (i) and (ii):

(i') *Ann knows that Bill has some way of committing suicide* ($\mathsf{K}_a\langle\langle b\rangle\rangle\Diamond\mathsf{suicide}$);
(ii') *Ann believes, taking only rational behaviour of all agents into account (in particular of Bill), that Bill has the ability to commit suicide* ($\mathsf{Bel}_a\langle\langle b\rangle\rangle\Diamond\mathsf{suicide}$).

Like before, (i') does not imply (ii'). While Ann knows that Bill "physically" has some way of killing himself, by assuming him to be rational she disregards

[2] This assumption is given in the plausibility model; it can be any assumption the designer would like to impose on the agents.

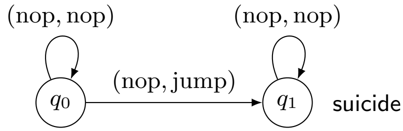



**Fig. 7.1.** A simple ICGSP.

the possibility. Bob's assumed rationality constrains his choices in Ann's view. This shows that in our logic knowing $\varphi$ does not imply rational belief in $\varphi$. We will justify the intuition in a more concrete example.

*Example 7.13.* There are two agents 1 (Ann) and 2 (Bill). Agent 2 has the ability to jump from a building and commit suicide. However, agent 1 disregards this possibility and considers it rational that 2 will not jump. The corresponding ICGSP is shown in Figure 7.1 where all different states are distinguishable from each other; the set of plausible strategy profiles consists of the single profile $s$ in which both agents play action *nop*, i.e., they do nothing (in particular, we want to impose that Bill does not jump). Hence, we have $\mathfrak{M}, q_0 \models \mathsf{K}_1\langle\langle 2\rangle\rangle \bigcirc \mathsf{suicide}$ but $\mathfrak{M}, q_0 \not\models \mathsf{Bel}_1\langle\langle 2\rangle\rangle \bigcirc \mathsf{suicide}$.

The following result, in line with [Bulling and Jamroga, 2007a], is immediate:

**Theorem 7.14.** *In general, standard (resp. constructive) knowledge does not imply standard (resp. constructive) rational belief. That is, formulae* $\mathbb{K}_a\varphi \wedge \neg\mathbb{B}\mathsf{el}_a\varphi$, $\mathbb{K}_a\varphi \wedge \neg\mathbb{W}_a\varphi$, $\mathsf{K}_a\varphi \wedge \neg\mathsf{Bel}_a\varphi$, $\mathsf{K}_a\varphi \wedge \neg\mathsf{W}_a\varphi$ *are satisfiable.*

*Proof.* The model from Example 7.13 can be used to show that all the formulas are satisfiable. ∎

## Non-Constructive Knowledge and Beliefs

In this section, we have a closer look at the standard (non-constructive) epistemic and doxastic operators. We mainly focus on strong beliefs; the cases for knowledge and weak beliefs are given analogously.

The non-constructive versions of distributed, common, and everybody belief are based on a specific construction involving the "until" operator. For example, the non-constructive belief of agent $a$ in $\varphi$, $\mathsf{Bel}_a\varphi$, is defined as $a$'s *constructive* belief in the ability of the empty coalition to enforce $\varphi$ until $\varphi$. In [Jamroga and Ågotnes, 2007] it was already shown that this definition captures the right notion; we recall the intuition here.

The cooperation modality $\langle\langle\emptyset\rangle\rangle$ ensures that the state formula $\varphi$ is evaluated *independently* in each indistinguishable state in $Q'$ (thus getting rid of its constructive flavour). However, a cooperation modality must be followed directly by a path formula, and $\varphi$ is a state formula. The trick is to use $\varphi\mathcal{U}\varphi$ instead, which ensures that $\varphi$ is true in the initial state of the path. Thus, $a$ believes in $\varphi$ iff $\mathbf{Pl}\,\varphi$ is independently true in every indistinguishable state. The following proposition (analogous to [Jamroga and Ågotnes, 2007, Theorem 46]) states that all non-constructive operators match their intended intuitions.

**Proposition 7.15.** *Let* $\mathfrak{M}$ *be an* ICGSP, $Q' \subseteq Q_{\mathfrak{M}}$, *and* $\varphi$ *be a* $\mathsf{CSLP}^{\mathsf{base}}$ *formula. Then, the following holds where* $\mathcal{K} = C, E, D$, *respectively:*

1. $\mathfrak{M}, Q' \models \mathcal{K}\mathsf{W}_A\varphi$ *iff* $\Upsilon = \emptyset \neq \mathbb{R}\mathrm{gt}$ *or* $\mathfrak{M}, q \models \varphi$ *for all* $q \in \mathrm{img}(Q', \sim_A^{\mathcal{K}})$*;*
2. $\mathfrak{M}, Q' \models \mathcal{K}\mathsf{Bel}_A\varphi$ *iff* $\mathfrak{M}, q \models \mathbf{Pl}\,\varphi$ *for all* $q \in \mathrm{img}(Q', \sim_A^{\mathcal{K}})$*;*
3. $\mathfrak{M}, Q' \models \mathcal{K}_A\varphi$ *iff* $\mathfrak{M}, q \models \mathbf{Ph}\,\varphi$ *for all* $q \in \mathrm{img}(Q', \sim_A^{\mathcal{K}})$*.*

*Proof.*

1. $\mathfrak{M}, Q' \models \mathcal{K}\mathsf{W}_A\varphi$ iff $\mathfrak{M}, \mathrm{img}(Q', \sim_A^{\mathcal{K}}) \models \langle\!\langle\emptyset\rangle\!\rangle \mathsf{Now}\varphi$ iff $\forall\lambda \in out(\mathrm{img}(Q', \sim_A^{\mathcal{K}}), s_\emptyset) : \mathfrak{M}, \lambda[0] \models \varphi$ iff $\Upsilon = \emptyset \neq \mathbb{R}\mathrm{gt}$ (cf. Remark 7.7) or $\forall q \in \mathrm{img}(Q', \sim_A^{\mathcal{K}}) : \mathfrak{M}, q \models \varphi$.
2. $\mathfrak{M}, Q' \models \mathcal{K}\mathsf{Bel}_A\varphi$ iff $\mathfrak{M}, Q' \models \mathbf{Ph}\,\mathcal{K}\mathbb{B}\mathsf{el}_A\langle\!\langle\emptyset\rangle\!\rangle \mathsf{Now}\mathbf{Pl}\,\varphi$ iff $\mathfrak{M}', \mathrm{img}(Q', \sim_A^{\mathcal{K}}) \models \langle\!\langle\emptyset\rangle\!\rangle \mathsf{Now}\mathbf{Pl}\,\varphi$ where $\mathbb{R}\mathrm{gt}_{\mathfrak{M}'} = \emptyset$ and otherwise $\mathfrak{M}'$ equals $\mathfrak{M}$ iff $\forall q \in \mathrm{img}(Q', \sim_A^{\mathcal{K}})$, $\mathfrak{M}', q \models \mathbf{Pl}\,\varphi$ iff $\forall q \in \mathrm{img}(Q', \sim_A^{\mathcal{K}})$, $\mathfrak{M}, q \models \mathbf{Pl}\,\varphi$.
3. Analogously.

■

### 7.1.5 The Full Logic **CSLP**

In this section we present $\mathcal{L}_{CSLP}$ which extends $\mathcal{L}_{CSLP}^{base}$ in such a way that plausibility terms are constructed from $\mathcal{L}_{CSLP}$-formulae. In the following we proceed in the very same way as in Sections 6.2.2 and Sections 6.2.3.

Analogously, to Definition 6.19 we define the languages $\mathcal{L}_{CSLP}^{k}$ and $\mathcal{L}_{CSLP}$.

**Definition 7.16 ($\mathcal{L}_{CSLP}^{k}$).** *The languages* $\mathcal{L}_{CSLP}^{k}(\Pi, \mathbb{A}\mathrm{gt}, \Omega)$ *and* $\mathcal{L}_{CSLP}(\Pi, \mathbb{A}\mathrm{gt}, \Omega)$ *are defined as in Definitions 6.19 and 6.20, respectively, but everywhere "ATLP" is replaced by "CSLP".*

The *extended plausibility mapping* is defined as in Definition 6.24 but the mapping is annotated with a set $Q'$ of states instead of a single state $q$.

An example $\mathcal{L}_{\mathrm{CSLP}}^{1}$ formula is

$$(\textbf{set-pl}\ \sigma.\langle\!\langle\emptyset\rangle\!\rangle\Box(\mathbf{Ph}\,\langle\!\langle\mathbb{A}\mathrm{gt}\rangle\!\rangle \bigcirc \mathsf{alive} \rightarrow (\textbf{set-pl}\ \sigma)\mathbf{Pl}\,\langle\!\langle\emptyset\rangle\!\rangle \bigcirc \mathsf{alive}))$$
$$\neg\mathsf{Bel}_a\langle\!\langle b\rangle\!\rangle\Diamond\mathsf{suicide}$$

which expresses the following: Assuming that rational agents avoid death whenever they can, it is not rational of Ann to believe that Bob can commit suicide.

Finally, we can define the semantics for $\mathcal{L}_{CSLP}$ and obtain the logic **CSLP** in the very same way as in Definition 6.26.

**Definition 7.17 (Semantics of $\mathcal{L}_{CSLP}$, CSLP).** *The semantics for $\mathcal{L}_{CSLP}$-formulae is given as in Definition 7.8 with the extended plausibility mapping $\widehat{[\![\cdot]\!]}_{\mathfrak{M}}$ used instead of $[\![\cdot]\!]_{\mathfrak{M}}$. I.e., only the semantic clauses for $(\textbf{set-pl}\,\omega)\varphi$ and $(\textbf{refn-pl}\,\omega)\varphi$ change as follows:*

$\mathfrak{M}, Q' \models (\textbf{set-pl}\,\omega)\varphi$ *iff* $\mathfrak{M}', Q' \models \varphi$ *where the new model* $\mathfrak{M}'$ *is equal to* $\mathfrak{M}$ *but the new set* $\Upsilon_{\mathfrak{M}'}$ *of plausible strategy profiles is set to* $\widehat{[\![\omega]\!]}^{Q'}_{\mathfrak{M}}$;

$\mathfrak{M}, Q' \models (\textbf{refn-pl}\,\omega)\varphi$ *iff* $\mathfrak{M}', Q' \models \varphi$ *where the new model* $\mathfrak{M}'$ *is equal to* $\mathfrak{M}$ *but the new set* $\Upsilon_{\mathfrak{M}'}$ *of plausible strategy profiles is set to* $\Upsilon_{\mathfrak{M}} \cap \widehat{[\![\omega]\!]}^{Q'}_{\mathfrak{M}}$.

*We have that* **CSLP** *is given by* $(\mathcal{L}_{CSLP}, \models)$.

## 7.2 Properties of CSLP

In this section, we examine the relationship between plausibility, knowledge and beliefs, and discuss the standard axioms about epistemic and doxastic concepts.

### 7.2.1 Plausibility, Knowledge and Beliefs

Firstly, we observe that knowledge is commutative with **Ph** and belief with **Pl**, which is a technically important property.

**Proposition 7.18.** *Let $\varphi$ be a $\mathcal{L}_{CSLP}$-formula. Then, we have that $\mathbf{Ph}\,\mathbb{K}_a\varphi \leftrightarrow \mathbb{K}_a\mathbf{Ph}\,\varphi$ and $\mathbf{Pl}\,\mathbb{B}\mathsf{el}_a\varphi \leftrightarrow \mathbb{B}\mathsf{el}_a\mathbf{Pl}$ are strongly valid.*

*Proof.* Both validities are clear from the following equations:

$$\mathbf{Ph}\,\mathbb{K}_a = \mathbf{Ph}\,(\mathbf{Ph}\,\mathbb{W}_a) = \mathbf{Ph}\,(\mathbf{Ph}\,\mathbb{W}_a)\mathbf{Ph} = (\mathbf{Ph}\,\mathbb{W}_a)\mathbf{Ph} = \mathbb{K}_a\mathbf{Ph}$$

and

$$\mathbf{Pl}\,\mathbb{B}\mathsf{el}_a = \mathbf{Pl}\,(\mathbb{W}_a\mathbf{Pl}) = \mathbf{Pl}\,(\mathbb{W}_a\mathbf{Pl})\mathbf{Pl} = (\mathbb{W}_a\mathbf{Pl})\mathbf{Pl} = \mathbb{B}\mathsf{el}_a\mathbf{Pl}.$$

■

From the definition of knowledge and belief it follows that a sequence of such operators collapses to the final operator in the sequence.

**Proposition 7.19.** *Let $a \in \mathbb{A}\mathrm{gt}$, $\varphi$ be a $\mathcal{L}_{CSLP}$-formula, and $X, Y$ be sequences of belief/knowledge operators; i.e. $X, Y \in \{\mathbb{B}\mathsf{el}_a, \mathbb{K}_a\}^*$. Then the following formulae are strongly valid:*

$$(i)\ X\mathbb{B}\mathsf{el}_a\varphi \leftrightarrow Y\mathbb{B}\mathsf{el}_a\varphi, \quad (ii)\ X\mathbb{K}_a\varphi \leftrightarrow Y\mathbb{K}_a\varphi.$$

*Proof.* The result is clear from the observation that the indistinguishability relations are equivalence relations and only the most recent plausibility operator is relevant. ■

In particular, we have that the following formulae are strongly valid: (1) $\mathbb{K}_a\mathbb{B}\mathsf{el}_a\varphi \leftrightarrow \mathbb{B}\mathsf{el}_a\varphi$: Agent $a$ knows that it believes $\varphi$ iff it believes $\varphi$; and (2) $\mathbb{B}\mathsf{el}_a\mathbb{K}_a\varphi \leftrightarrow \mathbb{K}_a\varphi$: Agent $a$ believes that it knows $\varphi$ iff it knows $\varphi$.

**Corollary 7.20.** *The following formulae are* not *valid: (i) $X\mathbb{B}\mathsf{el}_a\varphi \leftrightarrow Y\mathbb{K}_a\varphi$; (ii) $\mathbb{B}\mathsf{el}_a\varphi \rightarrow \mathbb{B}\mathsf{el}_a\mathbb{K}_a\varphi$; (iii) $\mathbb{B}\mathsf{el}_a\varphi \rightarrow \mathbb{K}_a\varphi$.*

*Proof.* (i) follows from (iii) and also from Theorem 7.14. (ii) follows from (iii) and Proposition 7.19. Example 7.13 provides a counterexample for (iii). We have that $\mathfrak{M}, q_0 \models \mathbb{B}\mathsf{el}_1\langle\langle 1\rangle\rangle \bigcirc \neg\mathsf{suicide}$ and $\mathfrak{M}, q_0 \not\models \mathbb{K}_1\langle\langle 1\rangle\rangle \bigcirc \neg\mathsf{suicide}$. ■

Corollary 7.20 expresses that (ii) an agent who has rational belief in $\varphi$ does not necessarily believe that it also knows $\varphi$; and (iii) an agent who believes in $\varphi$ does not necessarily know $\varphi$. Indeed, both formulae should intuitively not hold in a logics of knowledge and belief.

Our definitions of epistemic and doxastic operators from Section 7.1.2 strongly suggest that the underlying concepts are related. Let us consider formula $\mathbb{K}_a\mathbf{Pl}_B\varphi$: Agent $a$ has constructive knowledge in $\varphi$ if agents in $B$ behave rationally. This sounds similar to beliefs which is formally shown below.

**Proposition 7.21.** $\mathbf{Pl}_A\mathbb{K}_a\mathbf{Pl}_A\varphi \leftrightarrow \mathbf{Pl}_A\mathbb{W}_a\varphi$ *is strongly valid. We also have that $\mathbb{K}_a\varphi \leftrightarrow \mathbb{W}_a\varphi$ is valid (but not strongly valid).*

*Proof.* Unfolding $\mathbf{Pl}_A\mathbb{K}_a\mathbf{Pl}_A$ yields $\mathbf{Pl}_A\mathbf{Ph}\,\mathbb{W}_a\mathbf{Pl}_A$. It is easy to see that only the last plausibility operator is relevant which yields $\mathbb{W}_a\mathbf{Pl}_A$. Finally, it remains to observe that the plausibility operator is commutable with $\mathbb{W}_a$ as the latter does not effect the plausibility model.

The second claim follows since valid formulae are only interpreted in models in which all agents are assumed to play plausibly. The latter is not the case for strong validities, a counterexample is straightforward. ■

Finally, we conclude that rational beliefs and knowledge can also be defined in terms of each other.

**Theorem 7.22.** $\mathbb{B}\mathsf{el}_a\varphi \leftrightarrow \mathbb{K}_a\mathbf{Pl}\,\varphi$ *and* $\mathbb{K}_a\varphi \leftrightarrow \mathbb{B}\mathsf{el}_a\mathbf{Ph}\,\varphi$ *are strongly valid.*

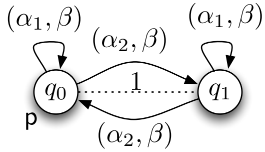



**Fig. 7.2.** ICGSP for the proof of Theorem 7.23.

*Proof.* Straightforward by unfolding the definitions and from the observation that only the last plausibility operator is relevant. ■

That is, believing in $\varphi$ is knowing that $\varphi$ plausibly holds, and knowing that $\varphi$ is believing that $\varphi$ is the case in all physically possible plays.

### 7.2.2 Axiomatic Properties

In this section we review the well-known **KDT45** axioms. For a modality $O$ these axioms are given as follows:

($\mathbf{K}_O$) $O(\varphi \to \psi) \to (O\varphi \to O\psi)$ ($\mathbf{D}_O$) $O\varphi \to \neg O \neg\varphi$
($\mathbf{T}_O$) $O\varphi \to \varphi$ ($\mathbf{4}_O$) $O\varphi \to OO\varphi$
($\mathbf{5}_O$) $\neg O\varphi \to O\neg O\varphi$

We say, for instance, that $O$ is an **K4** modality if axioms $\mathbf{K}_O$ and $\mathbf{4}_O$ are *strongly valid* over the class of all ICGSP. The following result is obtained in a way analogous to [Jamroga and Ågotnes, 2007, Theorem 37].

**Theorem 7.23 (Weak beliefs: KD45).** $\mathsf{W}_a$ *(standard weak beliefs) and* $\mathbb{W}_a$ *(constructive weak beliefs) are* ***KD45*** *modalities. Axiom* ***T*** *is not valid (resp. not strongly valid) for* $\mathbb{W}_a$ *(resp.* $\mathsf{W}_a$*).*

*Proof.*

**K** : Straightforward.
**D** : $\mathfrak{M}, Q' \models \mathbb{W}_a\varphi$ then $\mathfrak{M}, \mathrm{img}(Q', \sim_a) \models \varphi$ then not $\mathfrak{M}, \mathrm{img}(Q', \sim_a) \models \neg\varphi$ then not $\mathfrak{M}, Q' \models \mathbb{W}_a\neg\varphi$ then $\mathfrak{M}, Q' \models \neg\mathbb{W}_a\neg\varphi$.
**T** : Consider a model $\mathfrak{M}$ presented in Figure 7.2 taken from [Jamroga and Ågotnes, 2007]. We have that $\mathfrak{M}, q_0 \models \mathbb{W}_a\neg\langle\langle 1\rangle\rangle \bigcirc \mathsf{p}$ (since there is no uniform winning strategy) but not $\mathfrak{M}, q_0 \models \neg\langle\langle 1\rangle\rangle \bigcirc \mathsf{p}$. For $\mathsf{W}_a$ we consider the formula $\varphi = \langle\langle 1\rangle\rangle \bigcirc \mathsf{p}$ regarding the set $Q' = \{q_0, q_1\}$. Then, we have $\mathfrak{M}, Q' \models \mathsf{W}_a\langle\langle 1\rangle\rangle \bigcirc \mathsf{p}$ but $\mathfrak{M}, Q' \not\models \langle\langle 1\rangle\rangle \bigcirc \mathsf{p}$.
Axiom **T** is valid with respect to $\mathsf{W}_a$. $\mathfrak{M}, q \models \mathsf{W}_a\varphi$ iff $\mathfrak{M}, q \models \varphi$ for all $q \in \mathrm{img}(\{q\}, \sim_a^C)$ (by Proposition 7.15). The claim follows from the reflexivity of $\sim_a$.
**4** : Immediate from $\mathrm{img}(Q, \sim_a) = \mathrm{img}(\mathrm{img}(Q, \sim_a), \sim_a)$
**5** : Analogously to **D** and **4**.

■

*Remark 7.24.* Despite the similarities to [Bulling and Jamroga, 2007a], axiom **D** is not strongly valid for beliefs in **CTLKP** because the belief operator directly refers to plausible paths. Hence, if the set of paths is empty some formulae are trivially true ($\mathsf{Bel}\varphi$) and others are trivially false ($\neg\mathsf{Bel}\varphi$). In **CSLP** the notions of belief and plausibility are more modular.

For knowledge and strong beliefs it is easily seen that they also satisfy the **KD45** axioms. It just remains to check whether axiom **T** holds for knowledge or strong beliefs. However, for the same reason as in pure **CSL** this axiom does usually not hold; we refer to [Jamroga and Ågotnes, 2007] for a more detailled discussion on this issue. The problem that **T** is not true for knowledge (which is usually assumed to be a sensible requirement) is due to the non-standard semantics defined in terms of *sets* of states.

**Theorem 7.25 (Strong beliefs: KD45).** *Standard strong beliefs* $\mathsf{Bel}_a$ *and constructive strong beliefs* $\mathbb{B}\mathsf{el}_a$ *are* ***KD45*** *modalities. Axiom* ***T*** *is not valid for both notions of beliefs.*

*Proof.* The proof for **KD45** is done analogously to the one of Theorem 7.23. We proof that $\mathsf{Bel}_a$ is not valid. Again, we consider Example 7.13. Let $\omega$ be a plausibility term with $[\![\omega]\!]^{\{q_0\}} = \{s\}$ where $s$ is the strategy in which agents always perform the *nop* action. Then, we have that $\mathfrak{M}, q_0 \models \mathsf{Bel}_a(\textbf{set-pl } \omega)\langle\langle 1\rangle\rangle \bigcirc \neg\mathsf{suicide}$ but $\mathfrak{M}, q_0 \not\models (\textbf{set-pl } \omega)\langle\langle 1\rangle\rangle \bigcirc \neg\mathsf{suicide}$. In the latter case, no agent is assumed to play plausibly and all strategies are considered. ■

**Theorem 7.26 (Knowledge: KD45).** *Standard knowledge* $\mathsf{K}_a$ *and constructive knowledge* $\mathbb{K}_a$ *are* ***KD45*** *modalities. Axiom* ***T*** *is not valid for both notions of knowledge. Axiom* ***T*** *is not valid (resp. not strongly valid) for* $\mathbb{K}_a$ *(resp.* $\mathsf{K}_a$*).*

*Proof.* The proof for **KD45** is done analogously to the one of Theorem 7.23. Also the proof that **T** is not valid (resp. strongly valid) is done similarly. ■

If we consider a formula $\varphi$ which does not contain any constructive operators then the following holds.

**Theorem 7.27.** *Let* $\mathcal{L}$ *consist of all* **CSLP** *formulae that contain no constructive operators. Then:*

1. $\mathsf{K}_a$ *is a* ***KD45*** *modality in* $\mathcal{L}$*. Axiom* $\boldsymbol{T}_{\mathsf{K}_a}$ *is valid (but not strongly valid), and* $\mathsf{K}_a(\mathsf{K}_a\varphi \rightarrow \varphi)$ *is strongly valid in* $\mathcal{L}$*.*

*2.* $\mathsf{Bel}_a$ *is a* ***KD45*** *modality and* $\mathsf{Bel}_a(\mathsf{Bel}_a\varphi \to \varphi)$ *is strongly valid in* $\mathcal{L}$.

*Proof.*

1. That $\mathsf{K}_a$ is a **KD45** modality and that $\mathbf{T}_{\mathsf{K}_a}$ holds is shown in Theorem 7.25. For the latter part, we have $\mathfrak{M}, Q' \models \mathsf{K}_a(\mathsf{K}_a\varphi \to \varphi)$ iff $\mathfrak{M}', q \models \mathsf{K}_a\varphi \to \varphi$ for all $q \in img(Q', \sim_a)$ where $\mathfrak{M}'$ is the model in which no agent is supposed to play plausibly. Now, the result follows since there are no more constructive operators involved.
2. This part is shown analogously.

■

We observe that the validities $\mathsf{K}_a(\mathsf{K}_a\varphi \to \varphi)$ and $\mathsf{Bel}_a(\mathsf{Bel}_a\varphi \to \varphi)$ are similar to the truth axiom **T**.

### 7.2.3 Relationship to Existing Logics

In this section, we compare **CSLP** with some relevant logics and show their formal relationships. To this end, we use the notion of *embedding* from Definitions 6.28.

The following theorem is straightforward from the definition of the logic.

**Theorem 7.28.** **CSLP** *embeds* **ATL**, **ATLP**, *and* **CSL**.

It is easy to see that $\mathsf{K}_a$ is even a **KDT45** modality for the "non-strategic" sublanguage of $\mathcal{L}_{CSLP}$ and that this sublanguage can embed standard epistemic propositional logic.

**Proposition 7.29.** **CSLP** *embeds standard epistemic propositional logic.*

*Proof.* We define the sublanguage $\mathcal{L}$ of $\mathcal{L}_{CSLP}$ as the set of formulae generated by

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{K}_a\varphi$$

where $\mathsf{p} \in \Pi$ and $a \in \mathbb{A}\mathrm{gt}$. From Theorem 7.27 we know that $\mathsf{K}_a$ is a **KD45** modality. We show that axiom **T** is also strongly valid for $\mathcal{L}$. Let $\varphi_0$ be a propositional formula. Then, we have that $\mathfrak{M}, Q' \models \mathsf{K}_a\varphi_0$ iff $\forall q \in \mathrm{img}(Q', \sim_a)$, $\mathfrak{M}, q' \models \mathbf{Ph}\,\varphi_0$ iff $\forall q \in \mathrm{img}(Q', \sim_a)$, $\mathfrak{M}, q' \models \varphi_0$. This implies $\mathfrak{M}, Q' \models \varphi_0$ since $\varphi_0$ is purely propositional. The inductive step is done analogously. Finally, it is easy to see that $\mathcal{L}$ embeds standard epistemic logic. ■

The next result follows from Proposition 7.29 and Proposition 6.34.

**Proposition 7.30.** **CSLP** *embeds* **CTLKP** *in the class of epistemic Kripke structures.*

*Remark 7.31.* In [Jamroga and Ågotnes, 2007] and Section 6.3 it was shown that **CSL** and **ATLP**, respectively, embed several other logics, e.g., **ATEL** [van der Hoek and Wooldridge, 2003], **ATLI** [Jamroga et al., 2005], and **GLP** [van der Hoek et al., 2004]. Due to Theorem 7.28 all these logics are also embeddable in **CSLP**.

## 7.3 General Solution Concepts under Uncertainty

In [Jamroga and Bulling, 2007a; Bulling et al., 2009b] and Section 6.4 we have shown that **ATLP** can be used to reason about temporal properties of rational play. In particular it was shown that the logic allows to characterise game theoretic solution concepts of perfect information games [Osborne and Rubinstein, 1994]. These characterisations were then used to describe agents rational behaviour and impose the resulting rationality constraints on them. Here we show that **CSLP** can be used for the same purpose in the more general case of *imperfect information games* (IIGs). A natural question is how solution concepts for both game types differ?

Solution concepts for both kinds of games are very similar. For instance, a Nash equilibrium is a strategy profile from which no agent can deviate to obtain a better payoff, for both the perfect and imperfect information case. However, only *uniform strategies* are considered for IIG. Moreover, we require the agent to *know/identify* a strategy that is successful in *all* states indistinguishable for it.

In this section we characterise solution concepts for IIGs in $\mathcal{L}^1_{CSLP}$. Before we do that, however, we need some way to *evaluate* different strategies. We follow the approach of general solution concepts presented in Sections 3.3.2 and 6.4.2. So, we assume agents are equipped with *winning criteria* $\boldsymbol{\eta} = \langle \eta_1, \dots, \eta_k \rangle$ (one per agent) where $k = |\mathbb{A}\mathrm{gt}|$. Each criterion $\eta_a$ of agent $a$ is a temporal formula. Intuitively, a given strategy profile is successful for an agent $a$ iff the winning criterion is fulfilled on *all* resulting paths starting from *any* indistinguishable state given the strategy profile. This requirement is motivated by the fact that an agent does not know whether the system is in $q$ or $q'$ provided that $q$ and $q'$ are indistinguishable for it. The agent should play a strategy which is "good" in both states to ensure success. The following definition is the incomplete information counterpart of Definition 6.41.

**Definition 7.32 (Transform** ICGSP **to normal form game).** *Let $\mathfrak{M}$ be an* ICGSP*, $q \in Q_{\mathfrak{M}}$, and $\boldsymbol{\eta}$ be a vector of winning criteria.*

*We define $\mathcal{N}(\mathfrak{M}, \overrightarrow{\eta}, q)$, the* normal form game associated with *$\mathfrak{M}$, $\overrightarrow{\eta}$, and $q$, as the normal form game $\langle \mathbb{A}\mathrm{gt}, \mathcal{S}_1, \dots, \mathcal{S}_k, \mu \rangle$, where the set $\mathcal{S}_a$ of $a$'s strategies is given by $\Sigma^{ir}_a$ ($a$'s uniform strategies) for each $a \in \mathbb{A}\mathrm{gt}$, and the payoff function is defined as follows:*

$$\mu_a(a_1,\ldots,a_k) = \begin{cases} 1 & \textit{if } \mathfrak{M},\lambda \models \eta_a \\ & \textit{for all } \lambda \in out(img(q,\sim_a),\langle a_1,\ldots,a_k\rangle), \\ 0 & \textit{else.} \end{cases}$$

To give a clear meaning to solution concepts in an ICGSP, we relate them to the associated normal form game. The first solution concept we will define is a *best response strategy* for IIGs. Given a strategy profile $s_{-a} := (s_1, \ldots, s_{a-1}, s_{a+1}, \ldots, s_k)$ where $k = |\mathbb{A}\text{gt}|$, a strategy $s_a$ is said to be a *best response* to $s_{-a}$ if there is no better strategy for agent $a$ given $s_{-a}$. Now, $s$ is a *best response profile* wrt $a$ if $s_a$ is a best response against $s_{-a}$. According to 6.4.2 $\sigma$ is a best response profile for perfect information games wrt $a$ and $\boldsymbol{\gamma}$ in $\mathfrak{M}, q$ if

$$\mathfrak{M}, q \models (\textbf{set-pl } \sigma[\mathbb{A}\text{gt}\backslash\{a\}])\textbf{Pl}\left(\langle\!\langle a\rangle\!\rangle \eta_a \to (\textbf{set-pl } \sigma)\langle\!\langle\emptyset\rangle\!\rangle \eta_a\right).$$

The formula is read as follows: If agent $a$ has any strategy to enforce $\eta_a$ against $\sigma[\mathbb{A}\text{gt} \setminus \{a\}]$ then its strategy given by $\sigma$ should enforce $\eta_a$ as well.

What do we have to modify to make it suitable for imperfect information games? Firstly, we have to ensure that the strategy $\sigma$ is uniform, and indeed only uniform strategies are taken into account in the semantics of **CSLP**. Secondly, since the agent might not be aware of the real state of the system the described strategy should have its desired characteristics in each indistinguishable state. The agent should be able to *identify* the strategy; the main motivation of **CSL**. For this purpose **CSLP** provides the constructive belief operators; recall that $\mathbb{W}_a\langle\!\langle a\rangle\!\rangle$ means that $a$ has a single strategy successful in all indistinguishable states. To ensure this second point we just have to couple strategic operators with constructive operators. So we obtain the following description of a best response strategy for IIGs:

$$BR_a^{\boldsymbol{\eta}}(\sigma) \equiv (\textbf{set-pl } \sigma[\mathbb{A}\text{gt}\backslash\{a\}])\textbf{Pl}\,(\mathbb{W}_a\langle\!\langle a\rangle\!\rangle \eta_a \to (\textbf{set-pl } \sigma)\mathbb{W}_a\langle\!\langle\emptyset\rangle\!\rangle \eta_a).$$

Other solution concepts characterised in 6.4.2 can be adapted to IIGs following the same scheme, e.g.:

**Nash equilibrium (NE):** $NE^{\boldsymbol{\eta}}(\sigma) \equiv \bigwedge_{i\in\mathbb{A}\text{gt}} BR_i^{\boldsymbol{\eta}}(\sigma)$;
**Subgame perfect NE:** $SPN^{\boldsymbol{\eta}}(\sigma) \equiv \mathbb{E}\mathbb{W}_{\mathbb{A}\text{gt}}\langle\!\langle\emptyset\rangle\!\rangle\Box\, NE^{\boldsymbol{\eta}}(\sigma)$;
**Pareto optimal strategy (PO):**

$$PO^{\boldsymbol{\eta}}(\sigma) \equiv \forall\sigma'\ \textbf{Pl}\Big( \bigwedge_{a\in\mathbb{A}\text{gt}} ((\textbf{set-pl } \sigma')\mathbb{W}_a\langle\!\langle\emptyset\rangle\!\rangle \eta_a \to (\textbf{set-pl } \sigma)\mathbb{W}_a\langle\!\langle\emptyset\rangle\!\rangle \eta_a) \vee \bigvee_{a\in\mathbb{A}\text{gt}} ((\textbf{set-pl } \sigma)\mathbb{W}_a\langle\!\langle\emptyset\rangle\!\rangle \eta_a \wedge \neg(\textbf{set-pl } \sigma')\mathbb{W}_a\langle\!\langle\emptyset\rangle\!\rangle \eta_a\Big).$$

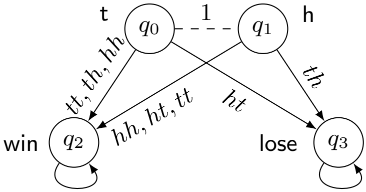
The following result shows that these concepts match their underlying intuitions.

**Theorem 7.33.** *Let* $\mathfrak{M}$ *be an* ICGSP, $q \in Q_{\mathfrak{M}}$, $\boldsymbol{\eta}$ *a vector of winning criteria, and* $\mathcal{N} := \mathcal{N}(\mathfrak{M}, \vec{\eta}, q)$*. Then, the following points hold:*

1. *The set of best response profiles wrt player* $a$ *in* $\mathcal{N}$ *is given by* $[\![\sigma.\widehat{BR^{\boldsymbol{\eta}}_a(\sigma)}]\!]^{\{q\}}_{\mathfrak{M}}$.
2. *The set of* NE *strategies in* $\mathcal{N}$ *is given by* $[\![\sigma.\widehat{NE^{\boldsymbol{\eta}}}(\sigma)]\!]^{\{q\}}_{\mathfrak{M}}$.
3. *The set of* PO *strategies in* $\mathcal{N}$ *is given by* $[\![\sigma.\widehat{PO^{\boldsymbol{\eta}}}(\sigma)]\!]^{\{q\}}_{\mathfrak{M}}$
4. *Let* $Q'$ *collect the states that any agent from* $\mathbb{A}\mathrm{gt}$ *considers possible (i.e.,* $\mathrm{img}(\{q\}, \sim^E_{\mathbb{A}\mathrm{gt}})$*) plus all states reachable from them by (a sequence of) temporal transitions. Then,* $[\![\sigma.\widehat{SPN^{\boldsymbol{\eta}}}(\sigma)]\!]^{\{q\}}_{\mathfrak{M}}$ *is equal to* $\bigcap_{q' \in Q'} [\![\sigma.\widehat{NE^{\boldsymbol{\eta}}}(\sigma)]\!]^{\{q'\}}_{\mathfrak{M}}$.

*Proof.* Let $[\![\sigma]\!]^{Q'} = s$ for all $Q' \subseteq Q$.

1. "$\subseteq$": Suppose $s_a$ is a best response to $s_{-a}$ in $\mathcal{N}$. Let $\omega$ be the plausibility term with denotation $s = (s_a, s_{-a})$. Then, if $\mu_a(s) = 0$ there is no other strategy $s'_a$ of $a$ such that $\mu_a(s'_a, s_{-a}) = 1$. Now, assume that $(\mathbf{set\text{-}pl}\ \sigma[\mathbb{A}\mathrm{gt}\backslash\{a\}])\mathbf{Pl}\,\mathbb{W}_a\langle\!\langle a\rangle\!\rangle\eta_a$ holds in $\mathfrak{M}, q$. Then, there is a strategy $s'_a$ of $a$ such that $\eta_a$ holds along all paths from $out(\mathrm{img}(a, \sim_a), (s'_a, s_{-a}))$; hence, $\mu_a(s'_a, s_{-a}) = 1$. Now, suppose that $s \notin [\![\sigma.\widehat{BR^{\boldsymbol{\eta}}_a(\sigma)}]\!]^{\{q\}}_{\mathfrak{M}}$; i.e., that $(\mathbf{set\text{-}pl}\ \sigma)\mathbb{W}_a\langle\!\langle\emptyset\rangle\!\rangle\eta_a$ does not hold in $\mathfrak{M}, q$. Then, there is a path in $out(\mathrm{img}(a, \sim_a), s)$ along which $\eta_a$ is false and thus $\mu_a(s) = 0$. Contradiction!
"$\supseteq$": Suppose $s \in [\![\sigma.\widehat{BR^{\boldsymbol{\eta}}_a(\sigma)}]\!]^{\{q\}}_{\mathfrak{M}}$ with $[\![\sigma]\!]^{\{q\}}_{\mathfrak{M}} = s$. That is, $BR^{\eta}_a(\sigma)$ is true in $\mathfrak{M}, q$. Then, following the same reasoning as above we have that if $\mu_a(s) = 0$; then, there is no other strategy $s'_a$ of $a$ such that $\mu_a(s'_a, s_{-a}) = 1$. I.e. $s_a$ is a best response to $s_{-a}$.
2. Follows from 1 and the fact that $s$ is a NE iff $s_a$ is a best response to $s_{-a}$ for each agent $a$.
3. "$\subseteq$": Let $s$ be Pareto optimal in $\mathcal{N}$. That is, there is no profile $s'$ such that for all agents $a$, $\mu_a(s') \geq \mu_a(s)$ and for some agent $a$, $\mu_a(s') > \mu_a(s)$. We show that $\mathfrak{M}, q \models PO^{\boldsymbol{\eta}}(\sigma)$. For the sake of contradiction assume the contrary; that is, $\mathfrak{M}, q \models \exists\sigma'\mathbf{Pl}\left(\bigvee_{a\in\mathbb{A}\mathrm{gt}}((\mathbf{set\text{-}pl}\ \sigma')\mathbb{W}_a\langle\!\langle\emptyset\rangle\!\rangle\eta_a \wedge \neg(\mathbf{set\text{-}pl}\ \sigma)\mathbb{W}_a\langle\!\langle\emptyset\rangle\!\rangle\eta_a) \wedge \bigwedge_{a\in\mathbb{A}\mathrm{gt}}((\mathbf{set\text{-}pl}\ \sigma)\mathbb{W}_a\langle\!\langle\emptyset\rangle\!\rangle\eta_a \rightarrow (\mathbf{set\text{-}pl}\ \sigma')\ \mathbb{W}_a\langle\!\langle\emptyset\rangle\!\rangle\eta_a))\right)$. We use $s'$ to denote $\sigma'$ for which the formula evaluates true. According to the left-hand site of the outermost (wrt. infix notation) conjunction, there has to be an agent $a'$ such that $\eta_a$ is achievable with respect to $s'$ but not with respect to $s$. From the right-hand side of the conjunction, we learn that the profile $s'$ is at least as good as $s$ (every payoff achievable following $s$ is also achievable following $s'$). However, this means that $s$ is not Pareto optimal. Contradiction!




| 1\2 | $h$ | $t$ |
|---|---|---|
| $h$ | 1, 1 | 0, 0 |
| $t$ | 0, 1 | 1, 1 |

**Fig. 7.3.** Simple ICGS.

"$\supseteq$": This part follows the same reasoning as the other direction.

4. We have that $s \in [\![\sigma.\widehat{SPN^{\eta}}(\sigma)]\!]_{\mathfrak{M}}^{\{q\}}$ iff for all paths $\lambda \in out(\mathrm{img}(q, \sim_{\mathbb{A}\mathrm{gt}}^{E}), s_{\emptyset})$ and all $i \in \mathbb{N}_0$, $\mathfrak{M}, \lambda[i, \infty] \models NE^{\eta}(\sigma)$ iff $\forall q' \in Q'$, $\mathfrak{M}, q' \models NE^{\eta}(\sigma)$ iff $\forall q' \in Q'$, $s \in [\![\sigma.\widehat{NE^{\eta}(\sigma)}]\!]_{\mathfrak{M}}^{\{q'\}}$ iff $s \in \bigcap_{q' \in Q'} [\![\sigma.\widehat{NE^{\eta}(\sigma)}]\!]_{\mathfrak{M}}^{\{q'\}}$.

■

*Example 7.34.* Let us consider the ICGSP shown in Figure 7.3. There are two agents, 1 and 2, and a coin which initially shows tail ($q_0$) or head ($q_1$); agent 1 cannot distinguish between them. Now, both agents win if 1 guesses the right side of the coin or if both agents agree on one side (regardless of whether it is the right one). For instance, the tuple $th$ denotes that 1 says tail and 2 head. Moreover, we assume that both agents have the winning criterion $\bigcirc$win. The associated NF game wrt $q_0$ is also given in Figure 7.3. Now we have that $[\![\sigma.\widehat{NE^{\eta}(\sigma)}]\!]_{\mathfrak{M}}^{\{q\}} = \{hh, tt\}$: Only if both agents agree on the same side, winning is guaranteed.

## 7.4 Uncertainty in Opponents' Behaviour

In this section we turn to another kind of incomplete information which is not related to information states provide to agents. We would like to recall the meaning of the cooperation modalities: $\langle\!\langle A \rangle\!\rangle \gamma$ is satisfied if the group of agents $A$ has a collective strategy to enforce temporal property $\gamma$. That is, $\langle\!\langle A \rangle\!\rangle \gamma$ holds if $A$ has a strategy that succeeds to make $\gamma$ true against *the worst* possible response from the opponents. So, the semantics of ATLs share the "all-or-nothing" attitude of many logical approaches to computation, justified by von Neumann's maximin evaluation of strategies in classical game theory [von Neumann and Morgenstern, 1944].

Such an assumption does seem appropriate in some application areas. For life-critical systems, security protocols, and expensive ventures like space missions it is indeed essential that nothing can go wrong (provided that the assumptions being made are correct). In many cases, however, one might be satisfied if the goal is achieved with reasonable likelihood. Also, it does not seem right to assume that the rest of agents will behave in the most hostile and destructive way; they may be friendly, indifferent, or simply not powerful enough (for example, due to incomplete knowledge). Thus, to evaluate available strategies, a finer measure of success is needed that takes into account the possibility of a non-adversary response.

A naive (but nevertheless appealing) idea is to evaluate a strategy $s$ by counting against how many opponents' responses it succeeds. If the ratio we get is, say 50%, we can say that $s$ succeeds in 50% of the cases. This approach is underpinned by the assumption that each response from the other agents is equally likely; that is, we in fact assume that those agents play at random. Or, putting it in another way: As we do not have any information about the future strategy of the opponents, we assume a uniform distribution over all possible response strategies. On the other hand, assuming the uniform distribution is too strong in many scenarios, where the "proponents" may have a more specific idea of what the opponents will do (obtained e.g. by statistical analysis and/or learning). In order to properly address the issue, we introduce modalities $\langle\langle A\rangle\rangle_{\omega}^{p}\gamma$ that express that *agents $A$ have a collective strategy to enforce $\gamma$ with probability of at least $p \in [0,1]$, assuming that the expected behaviour of the opponents is described by the prediction symbol $\omega$.*

In this section, we assume that the response from the opponents is independent from the actual strategy used by the proponents. It might be interesting to consider dependencies between choices of the two parties. This corresponds to situations in which the opponents have partial knowledge of the proponents' strategy.

We propose and discuss our new logics: ATLs *with probabilistic success.* Firstly, we define syntax and semantics on an abstract level. Then, we instantiate the semantics for two different ways of modelling the opponents' behaviour: mixed vs. behavioural memoryless strategies.

### 7.4.1 The Language $\mathcal{L}_{pATL}$

In $\mathcal{L}_{pATL}$, cooperation modalities $\langle\langle A\rangle\rangle$ of $\mathcal{L}_{ATL}$ are replaced with a richer family of strategic modalities $\langle\langle A\rangle\rangle_{\omega}^{p}$.

**Definition 7.35 ($\mathcal{L}_{pATL}$).** *The basic language $\mathcal{L}_{pATL}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ is defined over the nonempty sets $\Pi$ of propositions, $\mathbb{A}\mathrm{gt} = \{1, \dots, k\}$ of agents, and $\Omega$ of prediction symbols. The language consists of all* state formulae *$\varphi$ defined as follows:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle_{\omega}^{p}\gamma; \quad \textit{where} \quad \gamma ::= \bigcirc\varphi \mid \Box\varphi \mid \varphi\mathcal{U}\varphi,$$

*$\omega \in \Omega$, and $p \in [0,1]$. Additional temporal operators are defined as before.*

We use $\mathsf{p}, \omega, a, A$ to refer to a typical proposition, a prediction symbol, an agent, and a group of agents, respectively. The informal reading of formula $\langle\!\langle A \rangle\!\rangle_{\sigma}^{p}\gamma$ is: *Team $A$ can bring about $\gamma$ with success level of at least $p$ when the opponents behave according to $\sigma$.* The prediction symbols are used to assume some "predicted behaviour" of the opponents.

### 7.4.2 Semantics: The Generic Logic $\mathbf{pATL}_{\mathcal{BH}}$

In the following we define the semantics for $\mathcal{L}_{pATL}$ in a very generic way before considering more concrete settings. Models for $\mathbf{pATL}_{\mathcal{BH}}$ extend concurrent game structures with *strategic prediction denotation functions* (prediction denotation function for short) which assign prediction symbols to predicted behaviours of a given group of agents. For now, we use a non-empty set $\mathcal{BH}$ to refer to the agents possible *predicted behaviours*. There are several sensible ways how the set $\mathcal{BH}$ may actually be specified: *Mixed* and *behavioural strategies* provide two well-known possibilities. We will present the semantics for $\mathcal{L}_{pATL}$ based on these two notions in Sections 7.4.3 and 7.4.4, respectively. However, one could also think about other predictions, for instance, as a combination of mixed and behavioural strategies: The behaviour of some agents is predicted by the former and others by the latter.

**Definition 7.36 (Prediction denotation function).** *Let $\mathcal{BH}$ be a non-empty set representing possible (probabilistic) behaviours of the agents. A* prediction denotation function *is a function $[\![\cdot]\!] : \Omega \times \mathcal{P}(\mathbb{A}\mathrm{gt}) \to \mathcal{BH}$ where $[\![\omega, A]\!]$ denotes a (probabilistic) prediction of $A$'s behaviour according to the prediction symbol $\omega \in \mathfrak{Str}$. We write $[\![\omega]\!]_A$ for $[\![\omega, A]\!]$.*

Models for $\mathbf{pATL}_{\mathcal{BH}}$ extend CGSs with such functions.

**Definition 7.37 (pCGS).** *A* concurrent game structure with probability (pCGS) *is given by a tuple $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \Omega, [\![\cdot]\!] \rangle$ where $\langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o \rangle$ is a CGS, $\Omega$ is a set of probability terms, and $[\![\cdot]\!]$ is a prediction denotation function.*

The semantics of $\langle\!\langle A \rangle\!\rangle_{\omega}^{p}$ is based on a generic notion of a *success measure.* The actual instantiation of the notion will usually depend on a (probabilistic) prediction (from $\mathcal{BH}$) specified by the prediction denotation function and a prediction symbol. A success measure indicates "how successful" a group of agents is to enforce some property $\gamma$ (i.e. with which probability the formula may become satisfied) if the opponents behave according to their predicted behaviour.

**Definition 7.38 (Success measure).** *A* success measure *success is a function that takes a strategy (of the proponents) $s_A$, a probabilistic prediction*

$[\![\omega]\!]_{\mathbb{A}\mathrm{gt}\setminus A}$ *(of the opponents' behaviour), the current state of the system* $q$*, and the an* $\mathcal{L}_{pATL^\star}$*-formula* $\gamma$ *and returns a score* $success(s_A, [\![\omega]\!]_{\mathbb{A}\mathrm{gt}\setminus A}, q, \gamma) \in [0,1]$.

The semantics of **pATL**$_{\mathcal{BH}}$, parameterised by a success measure and a prediction denotation function, updates the **ATL** *Ir*-semantics from Definition 2.22 by replacing the rule for the cooperation modalities. Note, that we do only consider perfect information memoryless strategies for the proponents.

**Definition 7.39 (Semantics of pATL$_{\mathcal{BH}}$).** *Let* $\mathfrak{M}$ *be a* pCGS*. The semantics of* **pATL**$_{\mathcal{BH}}$ *updates the clauses from Definition 2.22 by replacing the clause for* $\langle\!\langle A\rangle\!\rangle$ *with the following:*

$\mathfrak{M}, q \models \langle\!\langle A\rangle\!\rangle^p_\omega \gamma$ *iff there is* $s_A \in \Sigma^{Ir}_A$ *such that* $success(s_A, [\![\omega]\!]_{\mathbb{A}\mathrm{gt}\setminus A}, q, \gamma) \geq p$.

Various success measures may prove appropriate for different purposes; they inherently depend on the type of the prediction denotation functions and therewith on the possible predicted behaviour represented by $\mathcal{BH}$.

### 7.4.3 Opponents' Play: Mixed Strategies

As the first instantiation of the generic framework, we consider *mixed memoryless strategies* which are probability distributions over pure memoryless strategies of the opponents. This notion of behaviour fits well our initial intuition of *counting the favourable opponents' responses* in order to determine the success level of a strategy.

**Definition 7.40 (Mixed memoryless strategy).** *A* mixed memoryless strategy *(*MMS*)* $\sigma_A$ *for* $A \subseteq \mathbb{A}\mathrm{gt}$ *is a probability measure over* $\mathcal{P}(\Sigma^{Ir}_A)$.

**Definition 7.41 (MMS denotation function).** *A* MMS denotation function *is a prediction denotation function with* $\mathcal{BH} = \bigcup_{A\subseteq \mathbb{A}\mathrm{gt}} \Xi(\Sigma^{Ir}_A)$*, such that* $[\![\omega]\!]_A \in \Xi(\Sigma^{Ir}_A)$. $[\![\omega]\!]_A(s)$ *denotes the probability that* $s$ *will be played by* $A$ *according to the prediction symbol* $\omega$.

Similarly, the abstract success measure introduced in Definition 7.38 can be instantiated as follows. A success measure for MMSs is given by a function which maps a strategy $s_A \in \Sigma^{Ir}_A$, a MMS $\sigma_{\mathbb{A}\mathrm{gt}\setminus A} \in \Xi(\Sigma_{\mathbb{A}\mathrm{gt}\setminus A})$, a state $q \in Q$, and an $\mathcal{L}_{pATL}$-path formula $\gamma$ to a value between 0 and 1, i.e. $success(s_A, \sigma_{\mathbb{A}\mathrm{gt}\setminus A}, q, \gamma) \in [0,1]$. The success function tells to what extent agents $A$ will achieve $\gamma$ by playing $s_A$ from $q$ on, when we expect the opponents ($\mathbb{A}\mathrm{gt} \setminus A$) to behave according to $\sigma_{\mathbb{A}\mathrm{gt}\setminus A}$.

In this paper, we take the success measure to be the *expected probability* of making $\gamma$ true. For this purpose, we first define the *outcome of a strategy*.

**Definition 7.42 (Outcome of a strategy against MMSs).** *The* outcome *of strategy $s_A$ against a mixed memoryless strategy $\sigma_{\mathbb{A}\text{gt}\backslash A}$ at state $q$ is the probability distribution over $\Lambda(q)$ given by:*

$$\mathcal{O}(s_A, \sigma_{\mathbb{A}\text{gt}\backslash A}, q)(\lambda) := \sum_{t \in Resp(s_A,\lambda)} \sigma_{\mathbb{A}\text{gt}\backslash A}(t)$$

*where $Resp(s_A, \lambda) = \{t \in \Sigma^{Ir}_{\mathbb{A}\text{gt}\backslash A} \mid \lambda \in out(q, \langle s_A, t\rangle)\}$ is the set consisting of all responses $t$ of the opponents that, together with $A$'s strategy $s_A$, result in path $\lambda$.*[3]

*Remark 7.43.* In Proposition 7.45 we show that the outcome is well defined.

Thus, $\mathcal{O}(s_A, \sigma_{\mathbb{A}\text{gt}\backslash A}, q)(\lambda)$ sums up the probabilities of all responses in $Resp(s_A, \lambda)$, for each path $\lambda$. As a consequence, $\mathcal{O}(s_A, \sigma_{\mathbb{A}\text{gt}\backslash A}, q)(\lambda)$ denotes the probability that the opponents will play a strategy resulting in $\lambda$. We also note again that, if memoryless strategies are played, the same action vector is performed every time a particular state is revisited, which restricts the set of paths that can occur.

**Definition 7.44 (Minimal periodic path, $\Lambda^{mp}(q)$).** *We say that a path $\lambda \in \Lambda(q)$ is* minimal periodic *if, and only if, the path can be written as $\lambda = \lambda[0, j]\lambda[j+1, i] \ldots \lambda[j+1, i]$ where $i \in \mathbb{N}_0$ is the minimal natural number such that there is some $j < i$ and $\lambda[i] = \lambda[j]$. The set of all minimal periodic paths starting in $q$ is denoted by $\Lambda^{mp}(q)$. For a finite model, the set $\Lambda^{mp}(q)$ consists of only finitely many paths.*

**Proposition 7.45.** *$\mathcal{O}(s_A, \sigma_{\mathbb{A}\text{gt}\backslash A}, q)$ is a probability measure over $\Lambda(q)$ and over $\Lambda^{mp}(q)$.*

*Proof.* That $\mathcal{O}(s_A, \cdot, q)$ is non-negative follows from the fact that $\sigma_{\mathbb{A}\text{gt}\backslash A}(t) \geq 0$ for all response strategies $t$. It is easy to see that all non minimal periodic paths have probability zero since they cannot occur of memoryless strategies are played. This implies that there are only finitely many paths with non-zero probability. Thus, $\mathcal{O}(s_A, \sigma_{\mathbb{A}\text{gt}\backslash A}, q)$ is $\sigma$-additive, and the following holds: $\mathcal{O}(s_A, \sigma_A, q)(\Lambda(q)) = \mathcal{O}(s_A, \sigma_A, q)(\Lambda^{mp}(q)) = \sum_{\lambda \in \Lambda^{mp}(q)} \sum_{t \in Resp(s_A,\lambda)} \sigma_B(t)$ $= \sum_{t \in \hat{Resp}(s_A)} \sigma_B(t)$ where $\hat{Resp}(s_A)$ consists of all strategies $t \in \Sigma^{Ir}_B$ such that there is a path $\lambda \in \Lambda^{mp}(q)$ with $\lambda \in out(q, \langle s_A, t\rangle)$. But then $\hat{Resp}(s_A) = \Sigma^{Ir}_B$ and thus the sum is equal to 1. ■

[3] Note that for a complete deterministic strategy profile $\langle s_A, t_{\mathbb{A}\text{gt}\backslash A}\rangle$ the outcome set contains exactly one path.

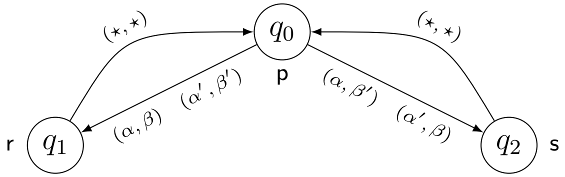



**Fig. 7.4.** A simple CGS $\mathfrak{M}_1 = \langle\{1,2\},\{q_0,q_1,q_2\},\{\mathsf{r},\mathsf{s}\},\pi,\{\alpha,\alpha',\beta,\beta'\},d,o\rangle$; $\pi$, $d$, and $o$ can be read off from the figure. By $\star$ we refer to any possible action.

**Definition 7.46 (Success measure with** MMSs**).** *The success measure for mixed memoryless strategies is defined as follows:*

$$success(s_A,\sigma_{\mathbb{A}\mathrm{gt}\setminus A},q,\gamma) = \sum_{\lambda\in\Lambda(q)} holds_\gamma(\lambda)\cdot\mathcal{O}(s_A,\sigma_{\mathbb{A}\mathrm{gt}\setminus A},q)(\lambda),$$

$$where\ holds_\gamma(\lambda) = \begin{cases} 1 & if\ \mathfrak{M},\lambda\models\gamma \\ 0 & else. \end{cases}$$

*Function $holds_\gamma : \Lambda \to \{0,1\}$ can be seen as characteristic function of the path formula $\gamma$: It indicates, for each path $\lambda$, whether $\gamma$ holds on $\lambda$.*

**Definition 7.47 ( $\mathbf{pATL}_{MMS}$).** *We define the logic $\mathbf{pATL}_{MMS}$ as the instantiation of $\mathbf{pATL}_{\mathcal{BH}}$ with the success measure and* MMS *denotation function as described in Definitions 7.46 and 7.41, respectively.*

By Proposition 7.45, $success(s_A,\sigma_{\mathbb{A}\mathrm{gt}\setminus A},q,\gamma)$ is indeed an expected value, and it is defined by a *finite* sum. Moreover, measuring the success of strategy $s_A$ by counting the favourable vs. all responses of the opponents is a special case, obtained by setting $[\![\omega]\!]_{\mathbb{A}\mathrm{gt}\setminus A}$ to the uniform probability distribution over $\Sigma_{\mathbb{A}\mathrm{gt}\setminus A}$.

*Example 7.48.* Let us consider a simple scenario with two agents 1 and 2 depicted in Figure 7.4. Agent 1 (resp. 2) can perform actions $\alpha$ and $\alpha'$ (resp. $\beta$ and $\beta'$). For example, strategy profile $(\alpha,\beta)$, performed in $q_0$, leads to state $q_1$ in which $\mathsf{r}$ holds. Agent 1 cannot enforce any of the outcomes on its own: $\mathfrak{M}_1,q_0 \models \neg\langle\langle 1\rangle\rangle\bigcirc\mathsf{r}\wedge\neg\langle\langle 1\rangle\rangle\bigcirc\mathsf{s}$, neither can agent 2. However, both agents have the power to determine the outcome when they cooperate: $\mathfrak{M}_1,q_0\models\langle\langle 1,2\rangle\rangle\bigcirc\mathsf{r}\wedge\langle\langle 1,2\rangle\rangle\bigcirc\mathsf{s}$.

It might be the case that additional information about 2's behaviour is available. Assume, for instance, that 1 has observed that 2 plays action $\beta'$ more often than $\beta$ (say, seven out of every ten times). This kind of observation can be formalised by a probability measure $\sigma$ over $\{\beta,\beta'\}$ with $\sigma_2(\beta)=0.3$ and $\sigma_2(\beta')=0.7$.

Using **ATL**, it was not possible to state any "positive" fact about 1's power. $\mathbf{pATL}_{MMS}$ allows a finer-grained analysis. We can now state that 1 can actually enforce any outcome (r or s) with probability at least 0.7. Formally, let $[\![\omega]\!]_2 = \sigma$. We have that $\mathfrak{M}, q_0 \models \langle\!\langle 1 \rangle\!\rangle_\omega^{0.7} \bigcirc \mathsf{r} \wedge \langle\!\langle 1 \rangle\!\rangle_\omega^{0.7} \bigcirc \mathsf{s}$. If 1 desires $r$, it should play $\alpha'$ since $\langle \alpha', \beta' \rangle$ leads to r; otherwise the agent should select action $\alpha$ in $q_0$.

### 7.4.4 Opponents' Play: Behavioural Strategies

In this section we present an alternative instantiation of the semantics, where the prediction of opponents' play is based on *behavioural strategies*. Such strategies are based on the Markovian assumption that the probability of taking an action depends only on the state where it is executed. We show that the semantics is well defined for $\mathcal{L}_{pATL}$.

**Definition 7.49 (Behavioural strategy).** *A* behavioural strategy *for* $A \subseteq \mathbb{A}\mathrm{gt}$ *is a function* $\beta_A : Q \to \bigcup_{q \in Q} \Xi(d_A(q))$ *such that* $\beta_A(q)$ *is a probability measure over* $d_A(q)$*, i.e.,* $\beta_A(q) \in \Xi(d_A(q))$*. We use* $\mathcal{B}_A$ *to denote the set of behavioural strategies of* $A$*.*

**Definition 7.50 (Behavioural strategy denotation function).**
*A* behavioural strategy denotation function *is a prediction denotation function with* $\mathcal{BH} = \bigcup_{A \subseteq \mathbb{A}\mathrm{gt}} \mathcal{B}_A$*, such that* $[\![\omega]\!]_A \in \mathcal{B}_A$*. Thus,* $[\![\omega_A]\!](q)(\overrightarrow{\alpha})$ *denotes the probability that the collective action* $\overrightarrow{\alpha}$ *will be played by agents* $A$ *in state* $q$ *according to the prediction symbol* $\omega$*.*

As in the case of mixed memoryless strategies (cf. Definition 7.42), the outcome of a strategy against behavioural predictions is a probability measure over paths. However, the setting is more complicated now. For mixed predictions it suffices to consider a probability distribution over the finite set of pure strategies which induces a probability measure over the set of paths. Indeed, only finite prefixes of paths, namely the non-looping parts, are relevant for the outcome (once a state is reentered, the same actions are performed again in a memoryless strategy). For behavioural strategies, actions (rather than strategies) are probabilistically determined, which makes it possible for different actions to be executed when the system returns to a previously visited state. Thus, the probability of a specific set of paths depends on the *complete* length of each path in the set, a finite prefix is not sufficient.

To define the outcome of a behavioural strategy we first need to define the probability space induced by the probabilities of one-step transitions. To this end, we follow the construction from [Kemeny et al., 1966]. We recall that $\Lambda(q)$ denotes the set of all infinite paths starting in $q$. The probability of a set of paths is defined inductively by consistently assigning probabilities to all finite initial segments (prefixes) of a path. The intuition is that prefix $h$ can be used to represent the set of infinite paths that extend $h$. By imposing the closure wrt complement and (countable) union, we obtain a probability measure over sets of paths.

We use $\Lambda^n(q)$ to denote the set of finite prefixes (histories) of length $n$ of the paths from $\Lambda(q)$. $\Lambda^n(q)$ is always finite for finite models. Now, we define $\mathcal{F}^n(q)$ and $\mathcal{F}(q)$ to be the following set of subsets of $\Lambda(q)$:

$$\mathcal{F}^n(q) := \big\{\{\lambda \mid \lambda[0, n-1] \in T\} \quad \big| \quad T \subseteq \Lambda^n(q)\big\} \quad \text{and} \quad \mathcal{F}(q) := \bigcup_{n=0}^{\infty} \mathcal{F}^n(q).$$

That is, for each set of prefixes $T \subseteq \Lambda^n(q)$, the set $\mathcal{F}^n(q)$ includes the set of all their infinite extensions. We note that each $\mathcal{F}^n(q)$ is a $\sigma$-algebra. Each element $S$ of $\mathcal{F}^n(q)$ (often called *cylinder set*) can be written as a finite union of *basic cylinder sets* $[h_i] := \{\lambda \in \Lambda(q) \mid h_i \leq \lambda\}$ where $h_i \in \Lambda^n(q)$ is a history of length $n$ and $h_i \leq \lambda$ denotes that $h_i$ is an initial prefix of $\lambda$. We have that $S = \bigcup_i [h_i]$ for appropriate $h_i \in \Lambda^n(q)$. We use these basic cylinder sets to define an appropriate probability measure.

A basic cylinder set $[h_i]$ consists of all extensions of $h_i$; hence, the probability that one of $h_i$'s extensions $\lambda \in [h_i]$ will occur is equal to the probability that $h_i$ will take place. Given a strategy $s_A$ and a behavioural response $\beta_{\mathbb{A}\mathrm{gt}\setminus A}$, the probability for $[h_i], h_i = q_0 \dots q_n$, is defined as the product of subsequent transition probabilities:

$$\nu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}([h_i]) := \prod_{i=0}^{n-1} \sum_{\overrightarrow{\alpha} \in Act(s_A, q_i, q_{i+1})} \beta_{\mathbb{A}\mathrm{gt}\setminus A}(q_i)(\overrightarrow{\alpha})$$

where $Act(s_A, q_i, q_{i+1}) = \{\overrightarrow{\alpha} \in d_{\mathbb{A}\mathrm{gt}\setminus A}(q_i) \mid q_{i+1} = o(q_i, \langle s_A(q_i), \overrightarrow{\alpha}\rangle)\}$ consists of all action profiles which can be performed in $q_i$ and which lead to $q_{i+1}$ given the choices $s_A$ of agents $A$. According to [Kemeny et al., 1966], the function $\nu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}$ is uniquely defined on $\mathcal{F}(q)$ and the restriction of $\nu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}$ to $\mathcal{F}^n(q)$ is a measure on $\mathcal{F}^n(q)$ for each $n$. It is also noted that $\mathcal{F}(q)$ is not a $\sigma$-algebra.

Therefore, we take $\mathcal{S}(q)$ to be the smallest $\sigma$-algebra containing $\mathcal{F}(q)$ and extend $\nu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}$ to a measure on $\mathcal{S}(q)$ as follows:

$$\mu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}(S) := \inf_{C \in \mathcal{H}(S)} \left\{ \sum_{[h] \in C} \nu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}([h]) \right\}$$

where $S \in \mathcal{S}(q)$ and $\mathcal{H}(S)$ denotes the denumerable set of coverings of $S$ by basic cylinder sets. That is, $\mathcal{H}(S)$ consists of sets $\{[h_1], [h_2], \dots\}$ such that $S \subseteq \bigcup_i [h_i]$. According to [Kemeny et al., 1966], we have that

$$(\Lambda(q), \mathcal{S}(q), \mu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}})$$

is a probability space. Actually, $\mu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}$ is the *unique* extension of $\nu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}$ on $\mathcal{F}^n(q)$ to the $\sigma$-algebra $\mathcal{S}(q)$ [Kemeny et al., 1966, Theorem 1.19]; in particular, this means that both measures coincide on all sets from $\mathcal{F}(q)$. We refer to $\mu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}$ as the *probability measure on* $\mathcal{S}(q)$ *induced by the pure strategy* $s_A$ *and the behavioural strategy* $\beta_{\mathbb{A}\mathrm{gt}\setminus A}$.

**Definition 7.51 (Success measure with behavioural memoryless strategies).** *As in the previous section, the success measure of strategy* $s_A$ *wrt the formula* $\gamma$ *is defined as the expected value of the characteristic function of* $\gamma$ *(i.e., holds*$_\gamma$*) over* $(\Lambda(q), \mathcal{S}(q), \mu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}})$.

$$success(s_A, \beta_{\mathbb{A}\mathrm{gt}\setminus A}, q, \gamma) := E[holds_\gamma] = \int_{\Lambda(q)} holds_\gamma d\mu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}.$$

Note that the formulation uses a *Lebesgue integral* over the $\sigma$-algebra $\mathcal{S}(q)$.

Now we can show that the semantics of $\mathcal{L}_{pATL}$ with behavioural strategies is well-defined. We first prove that $holds_\gamma$ is $\mathcal{S}(q)$-measurable (i.e., every preimage of $holds_\gamma$ is an element of $\mathcal{S}(q)$ and thus can be assigned a measure); then, we show that $holds_\gamma$ is integrable.

**Proposition 7.52.** *Function* $holds_\gamma$ *is* $\mathcal{S}(q)$*-measurable and* $\mu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}$*-integrable for any* $\mathcal{L}_{pATL}$*-path formula* $\gamma$.

The complete proof is given on page 306. This result allows to define the following logic.

**Definition 7.53 ($\mathsf{pATL}_{BS}$).** *We define the logic* $\mathsf{pATL}_{BS}$ *as the instantiation of* $\mathsf{pATL}_{\mathcal{BH}}$ *with the success measure and the* BS *denotation function as described in Definitions 7.51 and 7.50, respectively.*

Note that $\mathsf{pATL}_{BS}$ can be seen as a special case of the multi-agent Markov Temporal Logic MTL from [Jamroga, 2008a], since $\langle\langle A\rangle\rangle^p_\omega\gamma$ can be rewritten as the MTL formula $p \preccurlyeq (\mathbf{str}_{\mathbb{A}\mathrm{gt}\setminus A}\omega)\langle\langle A\rangle\rangle\gamma$.

## 7.5 Properties

Firstly, we observe that an analogous success measure can be constructed for **ATL**:

$$success_{\mathsf{ATL}}(s_A, q, \gamma) = \min_{\lambda \in out(s_A, q)} \{holds_\gamma(\lambda)\}.$$

Then,

$$\mathfrak{M}, q \models_{\textsf{ATL}} \langle\!\langle A \rangle\!\rangle \gamma \text{ iff there is } s_A \in \Sigma_A^{Ir} \text{ such that } success_{\textsf{ATL}}(s_A, q, \gamma) = 1.$$

Thus, the abstract framework can be instantiated in a way that embraces **ATL**. Alternatively, we can try to embed **ATL** in pATL using the probabilistic success measures we have already defined.

### 7.5.1 Embedding ATL in $\textsf{pATL}_{MMS}$

We consider $\mathcal{L}_{pATL}$ with mixed memoryless strategies. The idea is to require that every response strategy has a non-zero probability. Note that a given pCGS $\mathfrak{M}$ *induces* a CGS $\mathfrak{M}'$ in a straightforward way: Only the set of prediction symbols and the strategy denotation function must be left out. In the following we will also use pCGSs together with $\mathcal{L}_{ATL}$ formulae (without probability) by implicitly considering the induced CGSs.

**Theorem 7.54.** *Let $\gamma$ be a $\mathcal{L}_{ATL}$-path formula with no cooperation modalities, and let $\omega$ describe a mixed memoryless strategy symbol such that $[\![\omega]\!]_{\mathbb{A}\mathrm{gt}\setminus A}(t) > 0$ for every $t \in \Sigma_{\mathbb{A}\mathrm{gt}\setminus A}$. Then, for all models $\mathfrak{M}$ and states $q$ in $\mathfrak{M}$ it holds that:*

$$\mathfrak{M}, q \models_{\textsf{ATL}} \langle\!\langle A \rangle\!\rangle \gamma \textit{ iff } \mathfrak{M}, q \models_{\textsf{pATL}_{MMS}} \langle\!\langle A \rangle\!\rangle^1_\omega \gamma.$$

*Proof.* Let $\bar{A} := \mathbb{A}\mathrm{gt} \setminus A$ for $A \subseteq \mathbb{A}\mathrm{gt}$. "$\Rightarrow$": Assume that $s_A \in \Sigma_A^{Ir}$ and that for all $\lambda \in out(q, s_A)$ it holds that $\mathfrak{M}, \lambda \models \gamma$. Now suppose that $\mathfrak{M}, q \not\models_{\textsf{pATL}_{MMS}} \langle\!\langle A \rangle\!\rangle^1_\omega \gamma$. In particular that would mean that

$$success(s_A, \sigma_A, q, \gamma) = \sum_{\lambda \in \Lambda(q)} holds_\gamma(\lambda) \cdot \sum_{t \in Resp(s_A, \lambda)} \sigma_{\bar{A}}(t) < 1.$$

This can only be caused by two cases:
**(1)** There is a path $\lambda \in \Lambda(q)$ a strategy $t \in Resp(s_A, \lambda)$ with $\sigma_{\bar{A}}(t) > 0$ and $holds_\gamma(\lambda) = 0$. But then $\lambda \in out(q, s_A)$ contradicts the assumption that $s_A$ is successful.
**(2)** There is a strategy $t \in \Sigma_{\bar{A}}$ with $\sigma_{\bar{A}}(t) > 0$ and for all $\lambda \in \Lambda(q)$ it holds that $t \notin Resp(s_A, \lambda)$ (*). But there must be a path $\lambda$ with $\{\lambda\} = out(q, (s_A, t))$ and thus $t \in Resp(s_A, \lambda)$, which contradicts (*).

"$\Leftarrow$": Assume that $s_A \in \Sigma_A$ and $success(s_A, \sigma_A, q, \gamma) = 1$. Suppose that there is a path $\lambda \in out(q, s_A)$ with $\mathfrak{M}, \lambda \not\models \gamma$. This means that strategy $t$ with $out(q, (s_A, t)) = \{\lambda\}$ is in $Resp(s_A, \lambda)$ but plays no role in the calculation of the success value since $holds_\gamma(\lambda) = 0$. This contradicts the assumption that $success(s_A, \sigma_A, q, \gamma) = 1$. ■

The condition $[\![\omega]\!]_{\mathbb{A}\mathrm{gt}\setminus A}(t) > 0$ ensures that no "bad response" of the opponents is neglected because of zero probability. Since we only deal with finite models, the uniform distribution over $\Sigma_A^{Ir}$ is always well defined.

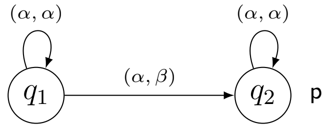



**Fig. 7.5.** CGS $\mathfrak{M}_2$ with actions $\alpha$ and $\beta$. The $\star \in \{\alpha, \beta\}$ refers to any of the two actions.

**Corollary 7.55.** *Let* $\mathsf{u_A}$ *be a term that denotes the uniform distribution over strategies of the agents in* $A$*, and let* $tr(\varphi)$ *replace all occurrences of* $\langle\langle A\rangle\rangle$ *by* $\langle\langle A\rangle\rangle^1_{\mathsf{u}_{\mathbb{A}\mathsf{gt}\setminus\mathsf{A}}}$ *in* $\varphi$*. Then,* $\mathfrak{M}, q \models_{\mathbf{ATL}} \varphi$ *iff* $\mathfrak{M}, q \models_{\mathbf{pATL}_{MMS}} tr(\varphi)$*.*

### 7.5.2 ATL versus pATL$_{BS}$

In this section we examine the connection between **ATL** and **pATL**$_{BS}$. In Theorem 7.54 we have shown that, under the semantics based on mixed response strategies, the **ATL** operator $\langle\langle A\rangle\rangle$ can be replaced by $\langle\langle A\rangle\rangle^1_\omega$ if all response strategies have non-zero probability according to $\omega$. One could expect the same for behavioural strategies if it is assumed that each "response action" is left possible; however, an analogous result does not hold.

The reason is that we consider probabilities over all infinite paths in the system, which necessitates a continuous probability space. Thus, the probability that a particular path will occur is zero. The following example shows that satisfying a path property with probability 1 does *not* imply that the property can be ensured in the sense of **ATL**.

*Example 7.56.* Let $\mathfrak{M}'_2$ be the pCGS based on CGS $\mathfrak{M}_2$ shown in Figure 7.5. We have that $\mathfrak{M}, q_1 \models \neg\langle\langle 1\rangle\rangle \Diamond \mathsf{p}$. What happens if agent 2 behaves according to a behavioural strategy? Let $\beta_2$ be the behavioural strategy specified as follows: $\beta_2(q_1)(\alpha) = \epsilon$, $\beta_2(q_1)(\beta) = 1 - \epsilon$, and $\beta_2(q_2)(\alpha) = 1$ where $0 < \epsilon < 1$. This behavioural strategy assigns non-zero probability to all actions of 2. Then, for a term $\omega$ with $[\![\omega]\!]_2 = \beta_2$ we have that $\mathfrak{M}, q_1 \models \langle\langle 1\rangle\rangle^1_\omega \Diamond \mathsf{p}$. Thus, 1 has a strategy which guarantees $\Diamond \mathsf{p}$ *with expected probability 1.* This is due to the fact that the only possible path which can prevent $\Diamond \mathsf{p}$ is $q_1 q_1 q_1 \ldots$. But the probability that this is going to happen is $\lim_{n\to\infty} \prod_{i=1}^n \epsilon = 0$. More formally, let $s_1$ be the strategy of 1 that assigns move $\alpha$ to be executed in both states. Then, we have

$$\begin{aligned}\int_{\Lambda(q)} holds_{\Diamond \mathsf{p}} d\mu^{s_1}_{\beta_2} &= \mu^{s_1}_{\beta_2}(holds_{\Diamond \mathsf{p}}) = 1 - \mu^{s_1}_{\beta_2}(\Lambda(q) \setminus holds_{\Diamond \mathsf{p}}) \\ &= 1 - \mu^{s_1}_{\beta_2}(\{(q_1)^\omega\}) = 1 - \mu^{s_1}_{\beta_2}(\bigcap_i [q_1^i]) = 1 - \lim_{i\to\infty} \mu^{s_1}_{\beta_2}([q_1^i]) \\ &= 1 - \lim_{i\to\infty} \epsilon^i = 1\end{aligned}$$

as $[q_1^i] \supset [q_1^{i+1}]$ for $i = 1, 2, \ldots$.

The example above shows that **pATL**$_{BS}$, if behavioural strategies are used, cannot simulate pure **ATL** operators in a straightforward way.

**Proposition 7.57.** *There is an $\mathcal{L}_{ATL}$-path formula $\gamma$, a model $\mathfrak{M}$ and a state $q$ such that $\mathfrak{M}, q \models_{\textsf{ATL}} \neg\langle\langle A\rangle\rangle\gamma$ but $\mathfrak{M}, q \models_{\textsf{pATL}_{BS}} \langle\langle A\rangle\rangle^1_\omega\gamma$ for every behavioural strategy $[\![\omega]\!]_{\mathbb{A}\mathrm{gt}\setminus A}$ in which every action is possible.*

*Proof.* A counter example is provided by Example 7.56. ■

Let us define a *sink state* as a state with a loop to itself being the only outgoing transition. A CGS (resp. pCGS) is *acyclic* iff it contains no cycles except for the loops at sink states. Such a model includes only a finite number of paths, so the following proposition can be proven analogously to Theorem 7.54.

**Proposition 7.58.** *Let $\mathfrak{M}$ be an acyclic* pCGS *and $\omega$ denote a behavioural prediction for $\mathbb{A}\mathrm{gt}\setminus A$ in which every action is possible (i.e., $[\![\omega]\!]_{\mathbb{A}\mathrm{gt}\setminus A}(q)(\overrightarrow{\alpha}) > 0$ for every $q \in Q$, $\overrightarrow{\alpha} \in d_{\mathbb{A}\mathrm{gt}\setminus A}(q)$). Then,*

$$\mathfrak{M}, q \models_{\textsf{ATL}} \langle\langle A\rangle\rangle\gamma \textit{ iff } \mathfrak{M}, q \models_{\textsf{pATL}_{BS}} \langle\langle A\rangle\rangle^1_\omega\gamma.$$

*Proof.* Let $q$ be some state in the model. If $q$ is a sink state, then there is exactly one path starting from it. In the case that $q$ is not a sink state there are only finitely many sequences of finite length to reach any sink state and clearly there is no path starting from $q$ without visiting a sink state. We have just argued that each acyclic model has only finitely many paths. Each finite sequence of states to a sink state has non-zero probability. Once a sink state is reached the probability of this sequence does not change. Hence, all possible paths have non-zero probability. The rest of the proof is done similarly to the proof of Theorem 7.54. ■

## 7.6 Summary

In this section we have proposed logics to model agents with incomplete information.

In the first setting, we have considered the (classical) indistinguishability between states. We have proposed the logic **CSLP** that relates epistemic and doxastic concepts in a specific way; more importantly, it allows to reason about the outcome of rational play in imperfect information games. In the logic beliefs are defined on top of the primitive notions of plausibility and indistinguishability. We have analysed the relationship between beliefs, knowledge, and rationality, and have proven in particular that rational beliefs form a **KD45** modality. **CSLP** embeds both **ATLP** and **CSL**; thus, the combination of knowledge, rationality, and strategic action turns out to be strictly more expressive than each of its parts if considered separately.

Moreover, we have shown how some important (generalised) solution concepts can be characterised and used for reasoning about imperfect information scenarios. In Section 10.3, we prove that the model checking problem for the basic variant of **CSLP** is $\mathbf{\Delta_3^P}$-complete. That is, the complexity of model checking is only slightly higher than for **CSL**, and no worse than for **ATLP**.

Secondly, we have considered a setting in which the opponents may be unable to identify the worst response to strategies of the proponents. We have combined the rigorous approach to success of ATLs with a quantitative analysis of the possible outcome of strategies. The resulting logic goes beyond the usual "all-or-nothing" reasoning: Instead of always looking at the opponents' worst response, we have assumed that they select strategies according to some probabilistic prediction. To this end, we have used new cooperation modalities $\langle\!\langle A \rangle\!\rangle_{\omega}^{p}\gamma$ with the intuitive reading that group $A$ has a strategy to enforce $\gamma$ with probability $p$ assuming that the opponents behave according to the probability distribution referred to by $\omega$. Although we have introduced two specific notions of success (one based on mixed response strategies, the other on behavioural predictions), the idea of success measure is generic and can be implemented according to the designer's needs. This enables the logic to be used in a very flexible way and in various scenarios.

We have shown that the semantics of **pATL**$_{MMS}$ based on mixed responses embeds **ATL**, while the semantics of **pATL**$_{BS}$ based on behavioural responses does not (or, at least, not in a straightforward way). Furthermore, in Section 10.4 we prove that model checking **pATL**$_{MMS}$ is significantly harder than for **pATL**$_{BS}$. Thus, we obtain the surprising result that the first semantics (which looks more intuitive and less mathematically advanced at first glance) turns out to be more difficult in terms of complexity when compared to the other semantics.

# 8

# Abilities of Resource-Bounded Agents

## Contents



The modelling and verification of multi-agent systems, in particular the *model checking problem* (i.e. whether a given property holds in a given model), have attracted much attention in recent years [Clarke et al., 1999; Clarke and Emerson, 1981; Alur et al., 2002; Kupferman et al., 2000; Pnueli and Rosner, 1989; Jamroga and Dix, 2008]. Most of these results focus on well-established logics like the computation tree logics or alternating time temporal logics [Clarke and Emerson, 1981; Alur et al., 2002]. Just recently these logics have been extended to verify various aspects of *rational* agents [Bulling et al., 2009b; Bulling and Jamroga, 2009c]. However, the basic idea of rational agents being autonomous entities perceiving changes in their environment and acting according to a set of rules or plans in the pursuit of goals does not take into account resources. However, many actions that an agent would execute in order to achieve a goal can – in real life – only be carried out in the presence of certain resources. Without sufficient resources some actions are not available, leading to plan failure. The analysis and verification of agent systems with resources of this kind is still in its infancy; the only work we are aware of in this direction is [Bulling and Farwer, 2010c,a; Alechina et al., 2009b,a, 2010].

In this chapter we take first steps in modelling resource bounded systems (which can be considered as the single agent case of the scenario just described) and resource-bounded multi-agent systems. Well-known computational models are combined with a notion of resource to enable a more systematic and rigorous specification and analysis of such systems. The main motivation of this chapter is to propose a fundamental formal setting.

For the single agent case, the proposed logic builds on *Computation Tree Logic* [Clarke and Emerson, 1981]. Essentially, the existential path quantifier $\mathsf{E}\varphi$ (there is a computation that satisfies $\varphi$) is replaced by $\langle\rho\rangle\gamma$ where $\rho$ represents a set of available resources. The intuitive reading of the formula is that there is a computation *feasible with the given resources* $\rho$ that satisfies $\gamma$.

In the multi-agent setting we extend $\mathcal{L}_{ATL}$ with resources, *resource-bounded agent logic*, and introduce restricted settings. The main motivation is to investigate the boundaries of what can and cannot be verified about resource-bounded agents. It turns out that the handling of resources is harder than it may seem at first glance: In Chapter 11 we prove that in many settings the model checking problem is undecidable. We do also consider the decidability of model checking for the single agent logics. We show that **RTL** (*resource-bounded tree logic*), the less expressive version, has a decidable model checking problem as well as restricted variants of the full logic **RTL**$^\star$ and its models.

## 8.1 Modelling Resource-Bounded Systems

In this section we introduce *resource-bounded models* (RBMs) for modelling systems with limited resources. Then, we propose the logics **RTL**$^\star$ and **RTL** (resource-bounded tree logics), for the verification of such systems. We introduce cover models and graphs and consider several properties and special cases of RBMs.

While most agent models do not come with an explicit notion of resources, there is some recent work that take resources into account. [Shaw et al., 2008] consider resources in conjunction with reasoning about an agent's goal-plan tree. Time, memory, and communication bounds are studied as resources in [Alechina et al., 2008]. In [Ågotnes and Walther, 2009] the abilities of agents under bounded memory are considered. In their setting, a winning strategy has to obey given memory limitations.

Moreover, we are interested in the reasoning about and modelling of abilities of *multiple* agents having limited resources at their disposal. In Section 8.4 we consider an extension of the resource-bounded setting introduced here in the context of multi-agent systems (influenced by **ATL** [Alur et al., 2002] a logic for reasoning about strategic abilities of agents). We show that the problem is undecidable in general. On the other hand, if productions of resources are not allowed (as in [Alechina et al., 2008]) it was recently shown that the model checking problem is decidable [Alechina et al., 2010]. The authors of [Alechina et al., 2010] do also propose a sound and complete axiomatisation of their resource-based extension of **ATL** (the logic is called *resource-bounded alternating-time temporal logic*).

### 8.1.1 Multisets

We define some variations of multisets used in the following sections. We assume that $\mathbb{N}_0 = \{0, 1, 2, \dots\}$ and $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$.

**Definition 8.1 ($\mathbb{Z}/\mathbb{Z}^\infty$-multiset,$X^{\pm}$, $X^{\pm}$, $\mathbb{N}_0/\mathbb{N}_0^\infty$-multiset, $X^{\oplus}$, $X^{\oplus}$ ).** *Let $X$ be a non-empty set.*

*(a) A* $\mathbb{Z}$-multiset $\mathbf{Z} : X \longrightarrow \mathbb{Z}$ *over the set $X$ is a mapping from the elements of $X$ to the integers.*
*A* $\mathbb{Z}^\infty$-multiset $\mathbf{Z} : X \longrightarrow \mathbb{Z} \cup \{-\infty, \infty\}$ *over the set $X$ is a mapping from the elements of $X$ to the integers extended by $-\infty$ and $\infty$.*
*The set of all* $\mathbb{Z}$-multisets *(resp.* $\mathbb{Z}^\infty$-multisets*) over $X$ is denoted by $X^{\pm}$ (resp. $X^{\pm}$).*

*(b) An* $\mathbb{N}_0$-multiset *(resp.* $\mathbb{N}_0^\infty$-multiset*)* $\mathbf{N}$ *over $X$ is a* $\mathbb{Z}$-multiset *(resp.* $\mathbb{Z}^\infty$-multiset*) over $X$ such that for each $x \in X$ we have $\mathbf{N}(x) \geq 0$. The set of all* $\mathbb{N}_0$-multisets *(resp.* $\mathbb{N}_0^\infty$-multisets*) over $X$ is denoted by $X^{\oplus}$ (resp. $X^{\oplus}$).*

Whenever we speak of a 'multiset' without further specification, the argument is supposed to hold for any variant from Def. 8.1. In general, we overload the standard set notation and use it also for multisets, i.e., $\subseteq$ denotes multiset inclusion, $\emptyset$ is the empty multiset, etc. We assume a global set of resource types $\mathcal{R}es$. The resources of an individual agent form a multiset over this set. $\mathbb{Z}$-multiset operations are straightforward extensions of $\mathbb{N}_0$-multiset operations.

Multisets are frequently written as formal sums, i.e., a multiset $\mathbf{M} : X \longrightarrow \mathbb{N}_0$ is written as $\sum_{x \in X} \mathbf{M}(x)$. Given two multisets $\mathbf{M} : X \longrightarrow \mathbb{N}_0$ and

$\mathbf{M}' : X \longrightarrow \mathbb{N}_0$ over the same set $X$, multiset union is denoted by $+$, and is defied as $(\mathbf{M} + \mathbf{M}')(x) := \mathbf{M}(x) + \mathbf{M}'(x)$ for all $x \in X$. Multiset difference is defined only if $\mathbf{M}$ has at least as many copies of each element as $\mathbf{M}'$. Then, $(\mathbf{M} - \mathbf{M}')(x) := \mathbf{M}(x) - \mathbf{M}'(x)$ for all $x \in X$. For $\mathbb{Z}$-multisets, $+$ is defined exactly as for multisets, but the condition is dropped for multiset difference, since for $\mathbb{Z}$-multisets negative multiplicities are possible. Finally, for $\mathbb{Z}^\infty$-multisets we assume the standard arithmetic rules for $-\infty$ and $\infty$ (for example, $x + \infty = \infty$, $x - \infty = -\infty$, etc).

We define multisets with a bound on the number of elements of each type.

**Definition 8.2 (Bounded multisets).** *Let $k, l \in \mathbb{Z}$. We say that a multiset $\mathbf{M}$ over a set $X$ is* $k$-bounded *iff $\forall x \in X$ $(\mathbf{M}(x) \leq k)$. We use ${}^kX^{\pm}_{\infty}$ to denote the set of all $k$-bounded $\mathbb{Z}^\infty$-multisets over $X$; and analogously for the other types of multisets.*

Finally, we define the (positive) restriction of a multiset with respect to another multiset, allowing us to focus on elements with a positive multiplicity.

**Definition 8.3 ((Positive) restriction, $\mathbf{M} \restriction_{\mathbf{N}}$).** *Let $\mathbf{M}$ be a multiset over $X$ and let $\mathbf{N}$ be a multiset over $Y$. The* (positive) restriction of $\mathbf{M}$ regarding $\mathbf{N}$, *$\mathbf{M} \restriction_{\mathbf{N}}$, is the multiset $\mathbf{M} \restriction_{\mathbf{N}}$ over $X \cup Y$ defined as follows:*

$$\mathbf{M} \restriction_{\mathbf{N}} (x) := \begin{cases} \mathbf{M}(x) & \textit{if } \mathbf{N}(x) \geq 0 \textit{ and } x \in Y, \\ 0 & \textit{otherwise.} \end{cases}$$

So, the multiset $\mathbf{M} \restriction_{\mathbf{N}}$ equals $\mathbf{M}$ for all elements contained in $\mathbf{N}$ which have a non-negative quantity, and 0 for all others elements.

### 8.1.2 Resource-Bounded Systems

A resource-bounded agent has at its disposal a (limited) repository of resources. Performing actions reduces some resources and may produce others; thus, an agent might not always be able to perform all of its available actions. In the single agent case considered here, this corresponds to the activation or deactivation of transitions.

**Definition 8.4 (Resources $\mathcal{R}es$, resource quantity (set), feasible).** *An element of the non-empty and finite set $\mathcal{R}es$ is called* resource. *A tuple $(r, c) \in \mathcal{R}es \times \mathbb{Z}^\infty$ is called* resource quantity *and we refer to $c$ as the* quantity *of $r$. A* resource-quantity set *is a $\mathbb{Z}^\infty$-multiset $\rho \in \mathcal{R}es^{\pm}$. Note that $\rho$ specifies a resource quantity for each $r \in \mathcal{R}es$.*

*Finally, a resource-quantity set $\rho$ is called* feasible *iff $\rho \in \mathcal{R}es^{\oplus}$; that is, if all resources have a non-negative quantity.*

We model resource-bounded systems by an extension of Kripke frames, allowing each transition to *consume* and *produce* resources. We assign pairs $(\mathbf{c}, \mathbf{p})$ of resource-quantity sets to each transition, denoting that a transition labelled $(\mathbf{c}, \mathbf{p})$ *produces* $\mathbf{p}$ and *consumes* $\mathbf{c}$.

**Definition 8.5 (Resource-bounded model).** *A* resource-bounded model *(*RBM*) is given by $\mathfrak{M} = (Q, \to, \Pi, \pi, \mathcal{R}es, t)$ where*

- *$Q$, $\mathcal{R}es$, and $\Pi$ are finite sets of states, resources, and propositions, respectively;*
- *$(Q, \to, \Pi, \pi)$ is a Kripke model; and*
- *$t : Q \times Q \to \mathcal{R}es^{\oplus} \times \mathcal{R}es^{\oplus}$ is a (partial)* resource function*, assigning to each transition (i.e., tuple $(q, q') \in \to$) a tuple of feasible resource-quantity sets. Instead of $t(q, q')$ we sometimes write $t_{q,q'}$ and for $t_{q,q'} = (\mathbf{c}, \mathbf{p})$ we use ${}^{\bullet}t_{q,q'}$ (resp. $t^{\bullet}_{q,q'}$) to refer to $\mathbf{c}$ (resp. $\mathbf{p}$).*

Hence, in order to make a transition from $q$ to $q'$, where $q \to q'$, the resources given in ${}^{\bullet}t_{q,q'}$ are *required*; and in turn, $t_{q,q'}{}^{\bullet}$ are *produced* after executing the transition. Note, that we only allow finite productions and consumptions.

A *path* of an RBM is a path of the underlying Kripke structure. We also use the other notions for paths introduced above.

The consumption and production of resources of a path can now be defined in terms of the consumptions and productions of the transitions it comprises. Intuitively, not every path of an RBM is feasible; consider, for instance, a system consisting of a single state $q$ only where $q \to q$ and $t^{\bullet}_{q,q} = {}^{\bullet}t_{q,q}$. It seems that the transition "comes for free" as it produces the resources it consumes; however, this is not the case. The path $qqq\dots$ is not feasible as the initial transition is not enabled due to the lack of initial resources. Hence, in order to enable it, at least the resources given in ${}^{\bullet}t_{q,q}$ are necessary. Intuitively, a path is said to be *$\rho$-feasible* if each transition in the sequence can be executed with the resources available at the time of execution.

**Definition 8.6 ($\rho$-feasible path, resource-extended path).** *A path $\lambda = q_1 q_2 q_3 \cdots \in \Lambda_{\mathfrak{M}}(q)$ where $q = q_1$ is called $\rho$-*feasible *if for all $i \in \mathbb{N}$ the resource-quantity set*

$$\left(\rho + \Sigma_{j=1}^{i-1}(t^{\bullet}_{q_j q_{j+1}} - {}^{\bullet}t_{q_j q_{j+1}})\right) \restriction_{{}^{\bullet}t_{q_i q_{i+1}}} - {}^{\bullet}t_{q_i q_{i+1}} \text{ is feasible.}$$

*A* resource-extended *path is given by $\lambda \in (Q \times \mathcal{R}es^{\pm})^{\omega}$ such that the restriction of $\lambda$ to states, denoted $\lambda|_Q$, is a path in the model and the second component keeps track of the currently available resources; we use $\lambda|_{\mathcal{R}es}$ to refer to the projection to the second component.*

### 8.1.3 Resource-Bounded Tree Logic

We present a logic based on $\mathbf{CTL}^\star$ which can be used to verify systems with limited resources. In the logic we replace the $\mathbf{CTL}^\star$ path quantifier $\mathsf{E}$ by $\langle\rho\rangle$ where $\rho$ is a resource-quantity set. The intuitive reading of a formula $\langle\rho\rangle\gamma$ is that there is a(n) (infinite) $\rho$-feasible path $\lambda$ on which $\gamma$ holds. Note that $\mathsf{E}$ (there is a path in the system) can be defined as $\langle\rho^\infty\rangle$ where $\rho^\infty$ is the resource set assigning $\infty$ to each resource type. Formally, the language is defined as follows.

**Definition 8.7 ($\mathcal{L}_{RTL^\star}$).** *Let $\mathcal{R}es$ be a set of resources. The language $\mathcal{L}_{RTL^\star}(\Pi, \mathcal{R}es)$ is defined by the following grammar:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\rho\rangle\gamma \text{ where } \gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \varphi\mathcal{U}\varphi \mid \bigcirc\varphi$$

*and $\mathsf{p} \in \Pi$ and $\rho \in \mathcal{R}es^{\pm}$. Formulae $\varphi$ (resp. $\gamma$) are called* state *(resp.* path*) formulae.*

*Moreover, we define fragments of $\mathcal{L}_{RTL^\star}$ in which the domain of $\rho$ is restricted. Let $X$ be any set of multisets over $\mathcal{R}es$. Then $\mathcal{L}_{RTL^\star_X}$ restricts $\mathcal{L}_{RTL^\star}$ in such a way that $\rho \in X$. Finally, we define $[\rho]$, the* dual of *$\langle\rho\rangle$, as $\neg\langle\rho\rangle\neg$.*

Analogously to the language of $\mathbf{CTL}$ we define $\mathcal{L}_{RTL}$ as the fragment of $\mathcal{L}_{RTL^\star}$ in which each temporal operator is immediately preceded by a path quantifier.

**Definition 8.8 ($\mathcal{L}_{RTL}$).** *Let $\mathcal{R}es$ be a set of resources. The language $\mathcal{L}_{RTL}(\Pi, \mathcal{R}es)$ is defined by the following grammar:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\rho\rangle \bigcirc \varphi \mid \langle\rho\rangle\Box\varphi \mid \langle\rho\rangle\varphi\mathcal{U}\varphi$$

*where $\mathsf{p} \in \Pi$, $\rho \in \mathcal{R}es^{\pm}$. Fragments $\mathbf{RTL}_X$ are defined in analogy to Def. 8.7.*

As in the language of $\mathbf{CTL}$ we define $\Diamond\varphi$ as $\top\mathcal{U}\varphi$ and we use the following abbreviations for the universal quantifiers (they are not definable as duals in $\mathcal{L}_{RTL}$ as, for example, $\neg\langle\rho\rangle\neg\Box\varphi$ is not an admissible $\mathcal{L}_{RTL}$-formula):

$[\rho] \bigcirc \varphi \equiv \neg\langle\rho\rangle \bigcirc \neg\varphi$,
$[\rho]\Box\varphi \equiv \neg\langle\rho\rangle\Diamond\neg\varphi$,
$[\rho]\varphi\mathcal{U}\psi \equiv \neg\langle\rho\rangle((\neg\psi)\mathcal{U}(\neg\varphi \wedge \neg\psi)) \wedge \neg\langle\rho\rangle\Box\neg\psi$.

Next, we give the semantics for both languages.

**Definition 8.9 (Semantics, $\mathbf{RTL}^\star$).** *Let $\mathfrak{M}$ be an* RBM, *let $q$ be a state in $\mathfrak{M}$, and let $\lambda \in \Lambda_{\mathfrak{M}}$. The semantics of $\mathcal{L}_{RTL^\star}$-formulae is given by the satisfaction relation $\models$ which is defined as follows:*

$\mathfrak{M}, q \models \mathsf{p}$ *iff* $\lambda[0] \in \pi(\mathsf{p})$ *and* $\mathsf{p} \in \Pi$*;*

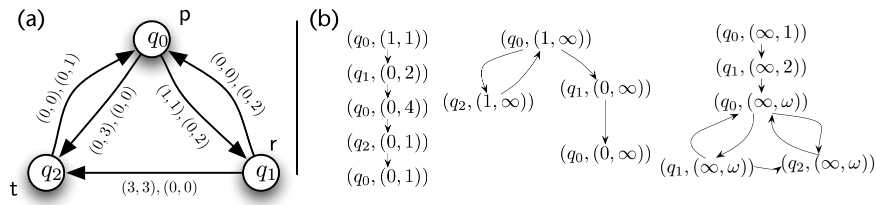



**Fig. 8.1.** In Figure (a) a simple RBM $\mathfrak{M}$ is shown and (b) presents some corresponding cover graphs.

$\mathfrak{M}, q \models \varphi \wedge \psi$ *iff* $\mathfrak{M}, q \models \varphi$ *and* $\mathfrak{M}, q \models \psi$*;*
$\mathfrak{M}, q \models \langle\rho\rangle\varphi$ *iff there is a* $\rho$*-feasible path* $\lambda \in \Lambda(q)$ *such that* $\mathfrak{M}, \lambda \models \varphi$*;*
$\mathfrak{M}, \lambda \models \varphi$ *iff* $\mathfrak{M}, \lambda[0] \models \varphi$*;*

*and for path formulae:*

$\mathfrak{M}, \lambda \models \neg\gamma$ *iff not* $\mathfrak{M}, \lambda \models \gamma$*;*
$\mathfrak{M}, \lambda \models \gamma \wedge \psi$ *iff* $\mathfrak{M}, \lambda \models \gamma$ *and* $\mathfrak{M}, \lambda \models \psi$*;*
$\mathfrak{M}, \lambda \models \Box\varphi$ *iff for all* $i \in \mathbb{N}$ *we have that* $\lambda[i,\infty] \models \varphi$*;*
$\mathfrak{M}, \lambda \models \bigcirc\varphi$ *iff* $\lambda[1,\infty] \models \varphi$*; and*
$\mathfrak{M}, \lambda \models \varphi\mathcal{U}\psi$ *iff there is an* $i \geq 0$ *such that* $\mathfrak{M}, \lambda[i,\infty] \models \psi$ *and* $\mathfrak{M}, \lambda[j,\infty] \models \varphi$ *for all* $0 \leq j < i$*.*

*We consider the logic* **RTL**$^\star$ *as the tuple* $(\mathcal{L}_{RTL^\star}, \models)$ *over all* RBMs *and analogously for all other fragments. These clauses are also used to define the semantics for* $\mathcal{L}_{RTL}$ *(therefore, we also stated the clause for* $\Box\varphi$*).*

Thus the meaning of $[\rho]\Box\mathsf{p}$ is that proposition $\mathsf{p}$ holds in every state on any $\rho$-feasible path.

We now discuss some interpretations of the formula $\langle\rho\rangle\gamma$ considering various resource-quantity sets. For $\rho \in \mathcal{R}es^{\oplus}$ it is assumed that $\rho$ consists of an initial (positive) amount of resources which can be used to achieve $\gamma$ where the quantity of each resource is finite. $\rho \in \mathcal{R}es^{\oplus}$ allows to *ignore* some resources (i.e., it is assumed that there is an infinite quantity of them). Initial debts of resources can be modelled by $\rho \in \mathcal{R}es^{\pm}$.

*Example 8.10.* Consider the RBM $\mathfrak{M}$ in Figure 8.1(a). Each transition is labeled by $(c_1, c_2), (p_1, p_2)$ with the interpretation: The transition consumes $c_i$ and produces $p_i$ quantities of resource $r_i$ for $i = 1, 2$. We encode the resource-quantity set by $(a_1, a_2)$ to express that there are $a_i$ quantities of resource $r_i$ for $i = 1, 2$.

- If there are infinitely many resources available proposition $\mathsf{t}$ can become true infinitely often: $\mathfrak{M}, q_0 \models \langle(\infty,\infty)\rangle\Box\Diamond\mathsf{t}$
- We have $\mathfrak{M}, q_0 \not\models \langle(1,1)\rangle\Box\top$ as there is no $(1,1)$-feasible path. The formula $\langle(1,\infty)\rangle\Box(\mathsf{p}\vee\mathsf{t})$ holds in $q_0$.
- Is there a way that the system runs forever given specific resources? Yes, if we assume, for instance, that there are infinitely many resources of $r_1$ and at least one resource of $r_2$: $\mathfrak{M}, q_0 \models \langle(\infty,1)\rangle\top$

These simple examples show, that it is not always immediate whether a formula is satisfied, sometimes a rather tedious calculation might be required.

## 8.2 Properties of Resource-Bounded Models

### 8.2.1 Cover Graphs and Cover Models

In this section we introduce a transformation of RBMs into Kripke models. This allows us, in general, to reduce truth in **RTL** to truth in **CTL** as shown in Section 8.3.1.

We say that a resource-quantity set *covers* another, if it has at least as many resources of each type with at least one amount actually exceeding that of the other resource-quantity set. We are interested in cycles of transition systems that produce more resources than they consume, thereby giving rise to unbounded resources of some type(s). This is captured by a *cover graph* for RBMs, extending ideas from [Karp and Miller, 1969] and requiring an ordering on resource quantities.

**Definition 8.11 (Resource ordering $<$).** *Let $\rho$ and $\rho'$ be resource sets in $\mathcal{R}es^{\pm}$. We say $\rho < \rho'$ iff $(\forall r \in \mathcal{R}es\ (\rho(r) \leq \rho'(r))) \wedge (\exists r \in \mathcal{R}es\ (\rho(r) < \rho'(r)))$. We say $\rho$ has strictly less resources than $\rho'$ or $\rho'$* covers *$\rho$.*

The ordering is extended to allow values of $\omega$ by defining for $x \in \mathbb{N}$ that $\infty + \omega = \infty$, $\infty - \omega = \infty$, $\omega - \infty = -\infty$, $\omega + x = \omega$, $\omega - x = \omega$, and $\omega < \infty$.

**Definition 8.12 ($\rho$-feasible transition, $\xrightarrow{\rho}$).** *We say that a transition $q \to q'$ is* $\rho$-feasible *and write $q\xrightarrow{\rho}q'$ if for all $r \in \mathcal{R}es$ we have that $0 < {}^{\bullet}t_{q,q'}(r)$ implies ${}^{\bullet}t_{q,q'}(r) \leq \rho(r)$.*

So, given a specific amount of resources $\rho$ a transition is said to be $\rho$-feasible if it can be traversed given $\rho$. A node of the cover graph consists of tuples $(q, (x_i)_{i=1,\dots,|\mathcal{R}es|})$ where $q$ is a state of the RBM and $(x_i)_i$ is a vector representing the currently available resources. The variable $x_i$ denotes that there are $x_i$ units of resource $r_i$.

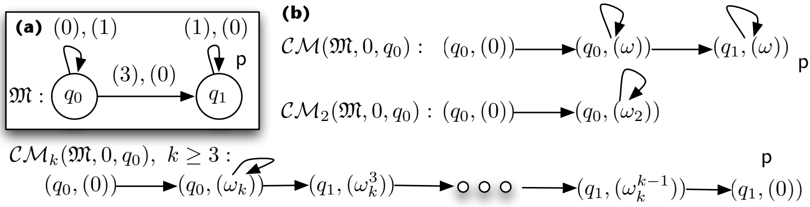



**Fig. 8.2.** An RBM $\mathfrak{M}$ (Fig. (a)), its cover model, 2-cover model, and $\kappa$-cover model (Fig. (b)).

**Definition 8.13 (($\rho, q$)-cover graph of an** RBM**, path, $\lambda|_Q$).** *Let $\mathfrak{M} = (Q, \to, \Pi, \pi, \mathcal{R}es, t)$, let $q$ be a state in $Q$, and let $\rho \in \mathcal{R}es^{\pm}$. Without loss of generality, assume $\mathcal{R}es = \{r_1, \dots, r_n\}$ and consider $(x_i)_i$ as an abbreviation for the sequence $(x_i)_{i=1,\dots,n}$. The $(\rho, q)$-*cover graph* $\mathcal{CG}(\mathfrak{M}, \rho, q)$ for $\mathfrak{M}$ with initial state $q \in Q$ and an initial resource-quantity set $\rho$ is the graph $(V, E)$ defined as the least fixed-point of the following specification:*

1. *$(q, (\rho(r_i))_i) \in V$ (the root vertex).*
2. *For $(q', (x_i)_i) \in V$ and $q'' \in Q$ with $q' \xrightarrow{(x_i)_i} q''$ then either:*
   *a) if there is a vertex $(q'', (\hat{x}_i)_i)$ on the path from the root to $(q', (x_i)_i)$ in $V$, with $(\hat{x}_i)_i < (x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i))_i$ then $(q'', (\tilde{x}_i)_i) \in V$ and $((q', (x_i)_i), (q'', (\tilde{x}_i)_i)) \in E$ where we define*

$$\tilde{x}_i := \begin{cases} \max\{\omega, x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i)\} & \text{if } \hat{x}_i < x_i, \\ x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i) & \text{otherwise;} \end{cases}$$

   *b) or else $(q'', (x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i))_i) \in V$ and $((q', (x_i)_i), (q'', (x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i))_i) \in E$.*

*A* path *in $\mathcal{CG}(\mathfrak{M}, \rho, q)$ is an infinite sequence of pairwise adjacent states. Given a path $\lambda = (q_1, (x_{1_i})_i)(q_2, (x_{2_i})_i)\dots$ we use $\lambda|_Q$ to denote the path $q_1 q_2 \dots$, i.e., the states of $\mathfrak{M}$ are extracted from the states in $V$.*

Cover graphs can be viewed as Kripke frames. It is obvious how they can be extended to models given an RBM.

**Definition 8.14 (($\rho, q$)-cover model of an** RBM**).** *Let $G = (V, E)$ be the $(\rho, q)$-cover graph of an* RBM *$\mathfrak{M} = (Q, \to, \Pi, \pi, \mathcal{R}es, t)$. The $(\rho, q)$-*cover model *of $\mathfrak{M}$, $\mathcal{CM}(\mathfrak{M}, \rho, q)$, is given by $(V, E, \Pi, \pi')$ with $\pi'((q, (x_i)_i)) := \pi(q)$ for all $(q, (x_i)_i) \in V$.*

Figure 8.2 shows the RBM $\mathfrak{M}$ in (a) and its cover model $\mathcal{CM}(\mathfrak{M}, 0, q_0)$ at the very top of (b). In the cover model, $\omega$ denotes the reachability of unbounded resources.

In Section 8.3.1 we analyse the relation between cover models and truth in **RTL**. Unfortunately, as illustrated in the next example, "simple" cover models in their current form are not yet suitable for that.

*Example 8.15.* Let $\lambda$ be the path of $\mathcal{CM}(\mathfrak{M}, 0, q_0)$ shown in Figure 8.2(b) with $\lambda|_Q = q_0 q_0 (q_1)^\omega$. Obviously, this path is not 0-feasible in model $\mathfrak{M}$ from Fig. 8.2(a). The problem is, that subsequent selections of the transition $q_0 \to q_0$ allows to generate any *finite* amount of resources, thus is covered by $\omega$, but any finite amount is not enough for the subpath $(q_1)^\omega$. This implies, that we cannot directly use cover models as alternative models.

We note, however, that the following result is obvious by the definition of a cover model: Each $\rho$-feasible path in the model is also a path in the corresponding cover model. The other direction is the one that causes trouble.

**Proposition 8.16.** *If $\lambda$ is a $\rho$-feasible $q$-path in $\mathfrak{M}$ then there is a $(q, \rho)$-path $\lambda'$ in $\mathcal{CM}(\mathfrak{M}, \rho, q)$ such that $\lambda = \lambda'|_Q$.*

*Proof.* Let $\lambda$ be a $\rho$-feasible $q$-path and $\eta_i$ be the resources available at $\lambda[i]$ after $\lambda[0, i]$, for $i = 1, 2, \ldots$; in particular, we have that $\eta_0 = \rho$. By induction on the number of transitions we show that there is a $(q, \rho)$-path $\lambda'$ in $\mathcal{CM}(\mathfrak{M}, \rho, q)$ where $(V, E)$ denotes the underlying graph such that $\lambda = \lambda'|_Q$. By definition is $(\lambda[0], \rho(r_i)_i) \in V$. For every state $q'$ with a $\rho$-feasible transition from $\lambda[0]$ to $q'$ we have that $(q', \ldots) \in V$ and an edge $((\lambda[0], \rho), (q', \ldots)) \in E$ (according to the construction of the cover model). In particular, we have that $(\lambda[1], \zeta) \in V$ with $\zeta \geq_\omega \eta_1$.

Now suppose the claim is proven up to position $k$. Let $(\lambda[k], \zeta) \in V$ with $\zeta \geq_\omega \eta_k$ be the $k + 1$st state on $\lambda'$. Following the same reasoning as above there is a transition $((\lambda[k], \zeta), (\lambda[k + 1], \zeta')) \in E$ with $\zeta' \geq_\omega \eta_{k+1}$. ■

In order to avoid the problem discussed in Example 8.15 we modify the cover graph construction as follows. The construction changes for those transitions that consume from the $\omega$ quantified resource type. Instead of using the rule "$\omega - k = \omega$", we (try to) expand the nodes for a fixed number of times ensuring that other loop's resource requirements can be met. But we abstain of introducing $\omega$'s as done in the cover graphs.

For the construction, we replace $\omega$ by $\kappa$ new symbols $\omega_\kappa^l$ for $l = 0, \ldots \kappa - 1$ and $\kappa \in \mathbb{N}_0$. For $i \in \mathbb{N}_0$ we define: $\omega_\kappa^l - i = 0$ for $l + i \geq \kappa$, $\omega_\kappa^l - i = \omega_\kappa^{l+i}$ for $l + i < \kappa$, $\omega_\kappa^l + i = \omega_\kappa^{\min\{l-i,0\}}$, and we set $\omega_\kappa = \omega_\kappa^0$. The symbol $\omega_\kappa$ is used to represent that at least $\kappa$ units of some resource type are stored, and $\omega_\kappa^l$ indicates that there are $\kappa - l$ resources left. Finally, we set $\omega_\kappa^l + \infty = \infty$.

Identifying the symbol $\omega_\kappa^l$ with the number $\kappa - l$ allows to extend the resource ordering from Definition 8.11 in a natural way; e.g. we have $i \leq \omega_\kappa^l$ iff $i \leq \kappa - l$. Moreover, this does also make it possible to lift the notation of

$\rho$-feasible transition etc. to this extended case. Finally, we define a class of cover models.

**Definition 8.17 ($\mathcal{CM}_\kappa(\mathfrak{M}, \rho, q)$).** *The construction of the* $(\rho, q, \kappa)$-cover graph *is defined as in Definition 8.13 but* $\omega$ *in 2. is replaced by* $\omega_\kappa$*; that is,*

$$\tilde{x}_i := \begin{cases} \max\{\omega_\kappa, x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i)\} & \text{if } \hat{x}_i < x_i, \\ x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i) & \text{otherwise.} \end{cases}$$

*The* $(\rho, q, \kappa)$-cover model*,* $\mathcal{CM}_\kappa(\mathfrak{M}, \rho, q)$*, is defined analogously to Definition 8.14.*

In Figure 8.2(b) we have also drawn the 2- and $\kappa$-cover model of the model $\mathfrak{M}$. In the next example we show that this generalised cover models overcome the problem discussed in Example 8.15.

*Example 8.18.* The "bad" path $\lambda$ of Example 8.15 is neither possible in $\mathcal{CM}_2(\mathfrak{M}, 0, q_0)$ nor in $\mathcal{CM}_\kappa(\mathfrak{M}, 0, q_0)$ for any $\kappa \geq 0$. This is, because for any fixed $\kappa$ the path $(q_1)^\omega$ will eventually have consumed all resources from $\omega_\kappa$.

However, another problem arises. If the $\kappa$ is chosen too small then we might abort the construction too early. The cover model $\mathcal{CM}_2(\mathfrak{M}, 0, q_0)$ illustrates the problem: In principal, it is possible to reach state $q_1$ if the loop $q_0 \to q_0$ is traversed at least three times. However, as $\omega_2$ does not allow to "remember" more than two units of resources state $q_1$ is never visited.

In order to avoid this problem we need to find an appropriate $\kappa$ such that a theorem similar to Proposition 8.16 with respect to $\kappa$-cover models holds. Indeed, such a $\kappa$ is constructible but the computation is computationally very expensive (cf. Theorem 11.9 and 11.13 and Remark 11.10).

We end the section with two results.

**Proposition 8.19.** *Let* $\rho \in \mathcal{R}es^\pm$*, let* $\mathfrak{M}$ *be an* RBM*, let* $q$ *be a state in* $\mathfrak{M}$*, and let* $G$ *denote the* $(\rho, q)$*- or* $(\rho, q, \kappa)$ *cover graph of* $\mathfrak{M}$*. Then, for each node* $(q, (x_i)_i)$ *of* $G$ *the property* $x_i \geq \min\{\rho(r_i), 0\}$ *holds.*

*Proof.* Suppose there is a node $(q, (x_i)_i)$ in the cover graph $G$ and an index $i$ such that $x_i < \min\{\rho(r_i), 0\}$. We first consider the case in which the minimum is equal to 0. Then, there must be a transition in $G$ which causes a non-negative quantity of $r_i$ to become negative. But such a transition is not feasible due to the construction of $G$! The case in which the minimum is equal to $\rho(r_i) < 0$ yields the same contradiction as a negative quantity of $r_i$ reduces even further which is not allowed in the construction of $G$. ■

The proposition states that non-positive resource quantities cannot decrease further. Theorem 8.20 states that cover models are finite; its proof is similar to the corresponding proof for Karp-Miller graphs [Karp and Miller, 1969].

**Theorem 8.20 (Finiteness of the ($\kappa$-)cover graph).** *Let $\rho \in \mathcal{R}es^{\pm}$ and $\kappa \in \mathbb{N}$. The $(\rho, q)$- and $(\rho, q, \kappa)$-cover graphs of the* RBM $\mathfrak{M}$, $q \in Q_{\mathfrak{M}}$, *are finite.*

*Proof.* Let $G$ denote the $(\rho, q)$-cover graph of $\mathfrak{M}$ and let $Q$ be the set of states in $\mathfrak{M}$. Assume $G$ is infinite (i.e., $G$ has infinitely many nodes). Then, there is an infinite path $l = v_1 v_2 \ldots$ in $G$ that contains infinitely many different states. Since $Q$ is finite there is some state, say $q' \in Q$, of $\mathfrak{M}$ and an infinite subsequence of distinct states $l' = v_{i_1} v_{i_2} \ldots$ on $l$ with $v_{i_j} = (q', (x_k^j)_k)$ and $i_j < i_{j+1}$ for all $j = 1, 2, \ldots$. Due to the construction of the cover graph, it cannot be the case that $(x_k^j)_k \leq (x_k^{j'})_k$ for any $1 \leq j < j'$; otherwise, an $\omega$-node would have been introduced and the infinite sequence would have collapsed. So, there must be two distinct indices, $o$ and $p$, with $1 \leq o, p \leq |\mathcal{R}es|$ such that, without loss of generality, $x_o^j < x_o^{j'}$ and $x_p^j > x_p^{j'}$. But by Prop. 8.19 we know that each $x_k^j \geq \min\{\rho(r_k), 0\}$; hence, the previous property cannot hold for all indices $o, p, j, j'$ but for the case in which $\rho(r) = -\infty$ for some resource $r$. However, this would also yield a contradiction as any non-negative resource quantity is bounded by 0. This proves that such an infinite path cannot exist and that the cover graph therefore has to be finite. ■

### 8.2.2 Resource-Bounded Models

In Section 11.2 we show that the model checking problem is decidable for **RTL**. Decidability of model checking for (full) **RTL**$^\star$ over arbitrary RBMs is still open. However, we identify interesting subclasses in which the problem is decidable. Below we consider some restrictions which may be imposed on RBMs.

**Definition 8.21 (Production free, zero (loop) free, $k$-bounded).** *Let $\mathfrak{M} = (Q, \rightarrow, \Pi, \pi, \mathcal{R}es, t)$ be an* RBM.

(a) *We say that $\mathfrak{M}$ is* production free *if for all $q, q' \in Q$ we have that $t_{q,q'} = (\mathbf{c}, \emptyset)$. That is, actions cannot produce resources they only consume them.*
(b) *We say that $\mathfrak{M}$ is* zero free *if there are no states $q, q' \in Q$ with $q \neq q'$ and $t_{q,q'} = (\emptyset, \mathbf{p})$. That is, there are no transitions between distinct states which do not consume any resources.*
(c) *We say that $\mathfrak{M}$ is* zero-loop free *if there are no states $q, q' \in Q$ with $t_{q,q'} = (\emptyset, \mathbf{p})$. That is, in addition to zero free models, loops without consumption of resources are also not allowed.*
(d) *We say that $\mathfrak{M}$ is (structurally) $k$-*bounded *for $\rho \in {}^k\mathcal{R}es_{\infty}^{\pm}$ iff the available resources after any finite prefix of a $\rho$-feasible path are bounded by $k$, i.e., there is no reachable state in which the agent can have more than $k$ resources of any resource type.*

In the following we summarise some results which are important for the model checking results presented in Section 11.2.

**Proposition 8.22.** *Let* $\mathfrak{M}$ *be an* RBM *and let* $\rho \in \mathcal{R}es^{\pm}$ *be a resource-quantity set. Then, there is an* RBM $\mathfrak{M}'$ *and a* $\rho' \in \mathcal{R}es^{\pm}$*, both effectively constructible from* $\mathfrak{M}$ *and* $\rho$*, such that the following holds: A path is* $\rho$*-feasible in* $\mathfrak{M}$ *if, and only if, it is* $\rho'$*-feasible in* $\mathfrak{M}'$*.*

*Proof.* Let $\rho'$ be equal to $\rho$ but the quantity of each resource $r$ with $\rho(r) \in \{-\infty, \infty\}$ is 0 in $\rho'$ and let $\mathfrak{M}'$ equal $\mathfrak{M}$ apart from the following exceptions. For each transition $(q, q')$ with $t_{qq'} = (\mathbf{c}, \mathbf{p})$ in $\mathfrak{M}$ do the following: Set $\mathbf{c}(r) = 0$ in $\mathfrak{M}'$ for each $r$ with $\rho(r) = \infty$; or remove the transition $(q, q')$ completely in $\mathfrak{M}'$ if $\mathbf{c}(r) > 0$ (in $\mathfrak{M}$) and $\rho(r) = -\infty$ for some resource $r$. Obviously, $\rho \in \mathcal{R}es^{\pm}$.

Now, the left-to-right direction of the result is straightforward as only transitions were omitted in $\mathfrak{M}'$ which can not occur on any $\rho$-feasible path in $\mathfrak{M}$. The right-to-left direction is also obvious as only resource quantities in $\mathfrak{M}'$ were set to 0 from which an infinite amount is available in $\rho$ and only those transitions were removed which can never occur due to an infinite debt of resources. ■

The next proposition presents some properties of special classes of RBMs introduced above. In general there may be infinitely many $\rho$-feasible paths. We study some restrictions of RBMs that reduce the number of paths:

**Proposition 8.23.** *Let* $\mathfrak{M} = (Q, \rightarrow, \Pi, \pi, \mathcal{R}es, t)$ *be an* RBM.

*(a) Let* $\rho \in \mathcal{R}es^{\pm}$ *and let* $\mathfrak{M}$ *be production and zero-loop free; then, there are no* $\rho$*-feasible paths.*
*(b) Let* $\rho \in \mathcal{R}es^{\pm}$ *and let* $\mathfrak{M}$ *be production and zero free. Then, for each* $\rho$*-feasible path* $\lambda$ *there is an (finite) initial segment* $\lambda'$ *of* $\lambda$ *and a state* $q \in Q$ *such that* $\lambda = \lambda' \circ qqq\ldots$*.*
*(c) Let* $\rho \in \mathcal{R}es^{\pm}$ *and let* $\mathfrak{M}$ *be production free. Then, each* $\rho$*-feasible path* $\lambda$ *has the form* $\lambda = \lambda_1 \circ \lambda_2$ *where* $\lambda_1$ *is a finite sequence of states and* $\lambda_2$ *is a path such that no transition in* $\lambda_2$ *consumes any resource.*
*(d) Let* $\rho \in \mathcal{R}es^{\pm}$ *and let* $\mathfrak{M}$ *be k-bounded for* $\rho$*. Then there are only finitely many state/resource combinations (i.e., elements of* $Q \times \mathcal{R}es^{\pm}$*) possible on any* $\rho$*-feasible path.*

*Proof.*

(a) As there are no resources with an infinite amount and each transition is production free and consumes resources some required resources must be exhausted after finitely many steps.

(b) Apart from (a) loops may come for free and this is the only way how $\rho$-feasible paths can result.

(c) Assume the contrary. Then, in any infinite suffix of a path there is a resource-consuming transition that occurs infinitely often (as there are only finitely many transitions). But then, as the model is production free, the path cannot be $\rho$-feasible.

(d) We show that there cannot be infinitely many state/resource combinations reachable on any $\rho$-feasible path. Since the condition of $\rho$-feasibility requires the consumed resources to be present, there is no possibility of infinite decreasing sequences of resource-quantity sets. This gives a lower bound for the initially available resources $\rho$. The $k$-boundedness also gives an upper bound. ■

We show that $k$-boundedness is decidable for RBMs.

**Proposition 8.24 (Decidability of $k$-boundedness).** *Given a model $\mathfrak{M}$ and an initial resource-quantity set $\rho$, the question whether $\mathfrak{M}$ is $k$-bounded for $\rho$ is decidable.*

*Proof.* First, we check that $\rho \in {}^k\mathcal{R}es^{\oplus}_{\infty}$. If this is not the case, then $\mathfrak{M}$ is not $k$-bounded for $\rho$. Then we construct the cover graph of $\mathfrak{M}$ and check whether there is a state $(q, (x_i)_i)$ in it so that $x_i > k$ for some $i$. If this is the case $\mathfrak{M}$ is not $k$-bounded; otherwise it is.

■

We end this section with a simple result showing a sufficient condition for a model to be $k$-bounded.

**Proposition 8.25.** *Let $\rho \in \mathcal{R}es^{\pm}$. Each production-free RBM is $k$-bounded for $\rho$ where $k := \max\{i \mid \exists r \in \mathcal{R}es\ (\rho(r) = i)\}$.*

## 8.3 Properties of Resource-Bounded Tree Logics

Before discussing specific properties of **RTL** and **RTL**$^\star$ and showing the decidability of the model checking problem for **RTL** and for special cases of **RTL**$^\star$ and its models, we note that our logics conservatively extend **CTL**$^\star$ and **CTL**. This is easily seen by defining the path quantifier $\mathsf{E}$ as $\langle\rho^\infty\rangle$ and by setting $t_{qq'} = (\emptyset, \emptyset)$ for all states $q$ and $q'$ where $\rho^\infty$ denotes the resource set assigning $\infty$ to each resource type. Hence, every Kripke model has a canonical representation as an RBM. Moreover, given an RBM we can express the existence of a path (neglecting resources) by $\mathsf{E} := \langle\rho^\infty\rangle$. This allows to directly interpret **CTL** and **CTL**$^\star$ formulae over RBMs.

**Proposition 8.26 (Expressiveness).** **CTL**$^\star$ *(resp.* **CTL***) can be embedded in* **RTL**$^\star$ *(resp.* **RTL***) over Kripke models and* RBMs.

*Proof.* Given a **CTL**$^\star$ formula $\varphi$ and a Kripke model $\mathfrak{M}$ we replace each existential path quantifier in $\varphi$ by $\langle\rho^\infty\rangle$ and denote the result by $\varphi'$. Then, we extend $\mathfrak{M}$ to the canonical RBM $\mathfrak{M}'$ if it is not already an RBM and have that $\mathfrak{M}, q \models \varphi$ iff $\mathfrak{M}', q \models \varphi'$. ■

### 8.3.1 RTL and Cover Models

We show that if there is a satisfying path in any $\kappa$-cover model; then, there also is a path in the corresponding RBM. Note however, that this result does only hold for *positive* formulae of the form $\langle\rho\rangle\gamma$.

Let $\lambda$ be a finite sequence of states. Then, we recursively define $\lambda^n$ for $n \in \mathbb{N}_0$ as follows: $\lambda^0 := \epsilon$ and $\lambda^i := \lambda^{i-1}\lambda$ for $i \geq 1$. That is, $\lambda^n$ is the path which results from putting $\lambda$ $n$-times in sequence.

The following lemma states that for flat $\mathcal{L}_{RTL}$-path formulae[1] it does not matter whether a cycle is traversed just once or many times. It can be proved by a simple induction on the path formula $\gamma$.

**Lemma 8.27.** *Let $\gamma$ be an $\mathcal{L}_{RTL}$-path formula containing no more path quantifiers, let $\mathfrak{M}$ be an* RBM *and let $\lambda$ be a path in $\mathfrak{M}$. Now, if $\tilde{\lambda} = q_1 \dots q_n$ is a finite subsequence of $\lambda$ with $q_1 = q_n$ (note, that a single state is permitted as well), then, $\lambda$ can be written as $\lambda_1\tilde{\lambda}\lambda_2$ where $\lambda_1, \lambda_2$ are subsequences of $\lambda$ and we have that : $\mathfrak{M}, \lambda \models \gamma$ if, and only if, $\mathfrak{M}, \lambda_1\tilde{\lambda}^n\lambda_2 \models \gamma$ for all $n \in \{1, 2, \dots\}$.*

The second lemma states that one can always extend a path in the $\kappa$-cover model to a feasible path in the RBM by duplicating loops.

**Lemma 8.28.** *Let $\lambda$ be a path in $\mathcal{CM}_\kappa(\mathfrak{M}, \rho, q), (q, \rho)$ and $\lambda' = \lambda|_Q$; then, there are tuples $(a_i, b_i, c_i) \in \mathbb{N}_0^2 \times \mathbb{N}$ for $i = 1, 2, \dots$ such that for all $j = 1, 2, \dots$ we have that $a_j \leq b_j < a_{j+1}$ and $\lambda'[a_j] = \lambda'[b_j]$ and the path*

$$(\lambda'[a_i, b_i]^{c_i})_{i=1,2,\dots} \text{ is } \rho\text{-feasible in } \mathfrak{M}.$$

*Proof.* Let a $(q, (\rho(r_i))_i)$-path $\lambda = l_1 l_2 \dots$ in $G := \mathcal{CM}_\kappa(\mathfrak{M}, \rho, q) = (V, E)$ be given. We extend $\lambda$ to a path $\lambda'$ (having the structure as stated in the lemma) such that $\lambda'|_Q$ is $\rho$-feasible in $\mathfrak{M}$.

If $\lambda|_Q$ is $\rho$-feasible we just take $\lambda'$ as $\lambda$. So, suppose $\lambda|_Q = q_{i_1} q_{i_2} \dots$ is not $\rho$-feasible. Then, there is a transition in $\lambda$ that is not feasible in $\mathfrak{M}$. Let $l_1 \dots l_{k+1}$ be the *minimal* length initial subpath of $\lambda$ such that $(l_1 \dots l_{k+1})|_Q$ is not feasible in $\mathfrak{M}$ and let $l_k = (q, (x_i)_i)$. According to the construction of cover graphs this can only be caused by a resource $r_l$ such that $x_l = \omega_\kappa^t$ for $0 \leq t \leq \kappa$. Let $l_o = (q', (x_i')_i)$ with $1 \leq o \leq k$ and $o$ maximal be the state on $\lambda$ at which $x_l'$ was set to $\omega_\kappa$ most recently. Then, there must be another state

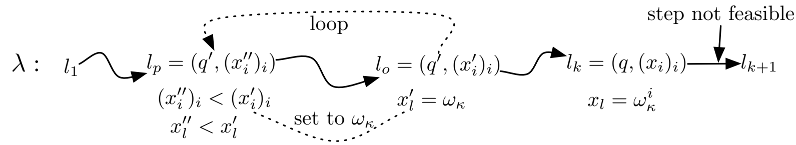



**Fig. 8.3.** Proof of Lemma 8.28.

$l_p = (q', (x''_i)_i)$, $1 \leq p < o$ and $p$ maximal, with $(x''_i)_i < (x'_i)_i$ and $x''_l < x'_l$. The setting is depicted in Figure 8.3.

So, we extend $\lambda$ to $\lambda'$ by duplicating the subsequence $l_p l_{p+1} \dots l_o$ in $l$ and adjusting the resources of the states preceding $l_p$ accordingly. Thus, we have that $\lambda'|_Q = q_{i_1} \dots q_{i_p} q_{i_{p+1}} \dots q_{i_o} q_{i_p} \dots q_{i_o} q_{i_{o+1}} \dots$. We subsequently continue this procedure (now applied to $\lambda'$) and do only duplicate transitions that are also present in $\lambda$ (i.e. not the new ones). It remains to show that this procedure does not force some $c_i$ to become infinite.

Suppose that there is some $c_i$ that becomes infinite following this construction. Then, there is a set of resources that requires the resources produced by $\lambda[a_i, b_i]$; and there is no other loop (or set of loops) that starts after $\lambda[b_i]$ that would also provide the needed resources (otherwise these loops would be duplicated as the construction looks for the latest possibility). In a $\kappa$-cover model, however, one can only "remember" $\kappa$ units of a resource; hence, one can have at most $\kappa$ transitions consuming of a specific resource until some other transition has to produce this very resource again. Thus, in order to ensure that $\lambda$ is a path in $G$ there must be a producing transition after $\lambda[b_i]$, in particular, a cycle introducing another $\omega_\kappa$-node following the same line of argumentation as above, which contradicts our supposition. Hence, we will actually obtain a path $\lambda'$ such that $\lambda'|_Q$ is $\rho$-feasible and has the structure $(\lambda|_Q[a_i, b_i]^{c_i})_{i=1,2,\dots}$. ■

**Theorem 8.29.** *Let $\rho \in \mathcal{R}es^{\pm}$, let $\mathfrak{M}$ be an* RBM, *let $q$ be a state in $\mathfrak{M}$. Then, for any $\kappa$ and any flat $\mathcal{L}_{RTL}$-formula $\langle\rho\rangle\gamma$ we have that:*

$$\textit{If } \mathcal{CM}_\kappa(\mathfrak{M}, \rho, q), (q, \rho) \models \mathsf{E}\gamma \textit{ then } \mathfrak{M}, q \models \langle\rho\rangle\gamma.$$

*Proof.* The result follows from Lemma 8.27 and 8.28. Firstly, the path $\lambda$ is extended to a path $\lambda'$ such that $\lambda'|_Q$ is $\rho$-feasible according to Lemma 8.28; then, Lemma 8.27 shows that the truth of the flat path formula according to $\lambda'$ does not change.

■

[1] A formula is said to be *flat* if it does not contain any path quantifier.

*Remark 8.30.* Note, that the proof of Theorem 11.13 gives an algorithm that particularly allows to construct a fixed index $\kappa$ from an RBM and $\langle\rho\rangle\gamma$ such that the "reverse" of Theorem 8.29 holds: If $\mathcal{CM}_\kappa(\mathfrak{M},\rho,q),(q,\rho) \models \mathsf{E}\gamma$ then $\mathfrak{M},q \models \langle\rho\rangle\gamma$. This construction of $\kappa$ however does already "solve" the model checking problem and is computationally very expensive.

### 8.3.2 RTL$^\star$ and Bounded Models

The case for **RTL**$^\star$ is more sophisticated as the language is able to characterise more complex temporal patterns. It is still open whether the general case is decidable. In the following, we discuss the effects of various properties of RBMs with respect to **RTL**$^\star$. For a given resource quantity it is possible to transform a structurally $k$-bounded RBM into a production-free RBM such that satisfaction of specific path formulae is preserved.

**Proposition 8.31.** *Let $\rho \in \mathcal{R}es^{\pm}$, let $\mathfrak{M}$ be a structurally $k$-bounded* RBM *for $\rho$, and let $q$ be a state in $\mathfrak{M}$. Then, we can construct a finite, production-free* RBM $\mathfrak{M}'$ *such that for every $\mathcal{L}_{RTL^\star}$-path formula $\gamma$ containing no more path quantifiers the following holds:*

$$\mathfrak{M}, q \models \langle\rho\rangle\gamma \quad \textit{if, and only if,} \quad \mathfrak{M}', q' \models \langle\emptyset\rangle\gamma.$$

*Proof.* We take $\mathfrak{M}'$ as the reachability graph of $\mathfrak{M}$. This graph is built similar to the cover graph but no $\omega$-nodes are introduced. Because there are only finitely many distinct state/resource combinations in $\mathfrak{M}$ (Prop. 8.23) the model is finite and obviously also production free.

Let $\mathfrak{M}, q \models \langle\rho\rangle\gamma$ and let $\lambda$ be a $\rho$-feasible path satisfying $\gamma$. Then, the path obtained from $\lambda$ by coupling each state with its available resources is a path in $\mathfrak{M}'$ satisfying $\gamma$. Conversely, let $\lambda$ be a path in $\mathfrak{M}'$ satisfying $\gamma$. Then, $\lambda|_Q$ is a $\gamma$ satisfying $\rho$-feasible path in $\mathfrak{M}$ due to the construction of $\mathfrak{M}'$. ■

The following corollary is needed for the model checking results in Section 11.2.

**Corollary 8.32.** *Let $\rho \in \mathcal{R}es^{\pm}$, let $\mathfrak{M}$ be a structurally $k$-bounded* RBM *for $\rho$, and let $q$ be a state in $\mathfrak{M}$. Then, we can construct a finite Kripke model such that for every $\mathcal{L}_{RTL^\star}$-path formula $\gamma$ containing no more path quantifiers the following holds:*

$$\mathfrak{M}, q \models \langle\rho\rangle\gamma \quad \textit{if, and only if,} \quad \mathfrak{M}', q' \models \mathsf{E}\gamma.$$

Lemma 8.33 states that loops that do not consume resources can be reduced to a fixed number of recurrences. For a path $\lambda$, we use $\lambda^{[n]}$ to denote the path which is equal to $\lambda$ but each subsequence of states $q_1q_2\ldots q_kq$ occurring in $\lambda$ with $q' := q_1 = q_2 = \cdots = q_k \neq q$ and $k > n$ where the transition $q' \to q'$

does not consume any resource (i.e. the first $k$ states represent a consumption-free loop that is traversed $k$ times) is replaced by $q_1q_2 \dots q_n q$. That is, states $q_{n+1}q_{n+2} \dots q_k$ are omitted. Note, that $\lambda^{[n]}$ is also well-defined for pure Kripke models. The following lemma follows as a special case from [Kucera and Strejcek, 2002].

**Lemma 8.33.**

(a) *Let $\mathfrak{M}$ be a Kripke model and $\gamma$ be a path formula of* $\mathsf{CTL}^\star$ *containing no path quantifiers and length $|\gamma| = n$. For each path $\lambda$ in $\Lambda_\mathfrak{M}$ we have that $\mathfrak{M}, \lambda \models \gamma$ if, and only if, $\mathfrak{M}, \lambda^{[n]} \models \gamma$ [Kucera and Strejcek, 2002].*

(b) *Let $\mathfrak{M}$ be a production- and zero-free* RBM *and $\gamma$ be an $\mathcal{L}_{RTL^\star}$-path formula containing no path quantifiers and length $|\gamma| = n$. Then, for each path $\lambda$ in $\Lambda_\mathfrak{M}$ the following holds true: $\mathfrak{M}, \lambda \models \gamma$ if, and only if, $\mathfrak{M}, \lambda^{[n]} \models \gamma$.*

Note that we might want to allow to re-enter loops $n$-times for cases in which the formula has the form $\bigcirc \bigcirc \dots \bigcirc \Diamond \varphi$.

## 8.4 Resource-Bounded Agent Logic

In this section we extend the single agent settings presented in the previous section to the multi-agent one.

*Resource-bounded tree logics*, introduced in [Bulling and Farwer, 2010a], extend the well-known computation tree logics [Clarke and Emerson, 1981] by resources. Instead of asking for the mere existence of an infinite path satisfying some temporal property, this path must also be feasible given a set of available resources. As shown in [Bulling and Farwer, 2010b] these logics can be considered as the resource-flat single agent fragments of the logics presented here.

*Resource-bounded coalition logic* (**RBCL**), an extension of coalition logic with resources, is introduced in [Alechina et al., 2009b]. This logic can be seen as a first step towards a multi-agent extension of the resource-bounded tree logics [Bulling and Farwer, 2010a] under the restricted temporal setting of multiple-step strategies ('sometime in the future'). Only recently, in [Alechina et al., 2010] a multi-agent version (**RBATL**) following the same ideas is presented. For both logics the authors allow only the consumption of resources which is computationally much easier and has a decidable model checking property (cf. Theorem 11.15). In [Bulling and Farwer, 2010b] we show that these logics can essentially be embedded in $\mathbf{RAL}_R$ (the perfect recall version).

**RBCL** is used in [Alechina et al., 2009a] to specify and verify properties about *coalitional resource games* [Wooldridge and Dunne, 2006]. These are games in which agents can cooperate and combine their available resources in order to bring about desired goals.

### 8.4.1 The Language $\mathcal{L}_{RAL^*}$

In this section we introduce the language **RAL**$^*$ (*resource-bounded agent logic*), *resource-bounded models* (RBAMs), and restricted variants of the logic. In the following we once more assume that $\mathcal{R}es = \{R_1, \ldots, R_u\}$ is a finite set of *resource types* or just *resources*.

We use an *endowment* function $\eta : \mathbb{A}\mathrm{gt} \times \mathcal{R}es \to \mathbb{N}_0^\infty$ to model the amount of resources an agent is equipped with[2]: $\eta(a, r)$ is the amount of resource $r$ agent $a$ possesses.

**Definition 8.34 (Endowment $\eta$, $\mathsf{En}$).** *An* endowment *is a function $\eta : \mathbb{A}\mathrm{gt} \times \mathcal{R}es \to \mathbb{N}_0^\infty$ to model the amount of resources an agent is equipped with. The set of all endowments is denoted by $\mathsf{En}$. We also write $\eta_a$ for $\eta(a)$.*

The quantity "$\infty$" is used to equip an agent with an infinite amount of resources. This allows us to ignore specific resource types for that agent. We define the endowment $\eta^\infty$ as the constant function that maps every resource for every agent to $\infty$. Finally, we use a *resource-quantity mapping* (RQM) $\rho : \mathcal{R}es \to \mathbb{Z}_\infty$ to model the currently available resources (in the system); that is, $\rho(r)$ denotes to availability or lack of resource $r$.

**Definition 8.35 (Language $\mathcal{L}_{RAL^*}$).** *The language $\mathcal{L}_{RAL^*}(\Pi, \mathbb{A}\mathrm{gt}, \mathsf{En})$ is defined as follows:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle_B \gamma \mid \langle\langle A \rangle\rangle_B^\eta \gamma$$

*where*

$$\gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \varphi \mathcal{U} \varphi \mid \bigcirc \varphi,$$

*$A, B \subseteq \mathbb{A}\mathrm{gt}$, $\mathsf{p} \in \Pi$, and*[3] *$\eta \in \mathsf{En}$. Formula $\varphi$ (resp. $\gamma$) is called* state *formula (resp.* path *formula). Moreover, we use $\langle\langle A \rangle\rangle^\eta$ (resp. $\langle\langle A \rangle\rangle$) as an abbreviation for $\langle\langle A \rangle\rangle_A^\eta$ (resp. $\langle\langle A \rangle\rangle_A$).*

The temporal operators $\bigcirc$ and $\mathcal{U}$ have their standard meaning *'in the next moment'* and *'until'*, respectively. As usual, one defines $\Diamond\gamma \equiv \top\mathcal{U}\gamma$ (*eventually*) and $\Box\gamma = \neg\Diamond\neg\gamma$ (*always from now on*).

### 8.4.2 The Semantics

As models for our logic we take CGSs and extend them by resources and a mapping $t$ indicating how many resources each action requires or produces when executed.

[2] $\mathbb{N}_0^\infty$ (resp. $\mathbb{Z}_\infty$) is defined as $\mathbb{N}_0 \cup \{\infty\}$ (resp. $\mathbb{Z} \cup \{\infty\}$).

[3] As we are mainly interested in decidability results about this logic the concrete representation of $\eta$ is irrelevant.

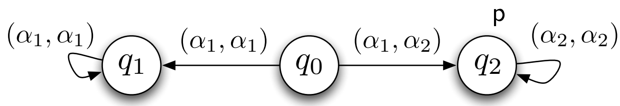



**Fig. 8.4.** A simple RBAM.

**Definition 8.36 (RBAM).** *A* resource-bounded (agent) model *(*RBAM*) is given by*

$$\mathfrak{M} = (\mathbb{A}\text{gt}, Q, \Pi, \pi, Act, d, o, \mathcal{R}es, t)$$

*where* $(\mathbb{A}\text{gt}, Q, \Pi, \pi, Act, d, o)$ *is a* CGS *and the function*

$$t : Act \times \mathcal{R}es \rightarrow \mathbb{Z}$$

*models the resources consumed and produced by actions. We define* $\mathsf{prod}(\alpha, r) := \max\{0, t(\alpha, r)\}$ *(resp.* $\mathsf{cons}(\alpha, r) := -\min\{0, t(\alpha, r)\}$*) as the amount of resource* $r$ *produced (resp. consumed) by action* $\alpha$*.*

*For* $\boldsymbol{\alpha} = \langle \alpha_1, \dots, \alpha_k \rangle$*, we use* $\boldsymbol{\alpha}|_A$ *to denote the sub-tuple consisting of the actions of agents* $A \subseteq \mathbb{A}\text{gt}$ *and we use* $X_{\mathfrak{M}}$ *to refer to an element* $X$ *contained in* $\mathfrak{M}$*.*

*Example 8.37.* In Figure 8.4 a simple RBAM $\mathfrak{M}$ with $\mathbb{A}\text{gt} = \{1, 2\}$, $d_1(q_0) = d_1(q_1) = d_2(q_1) = \{\alpha_1\}$, $d_1(q_2) = d_2(q_2) = \{\alpha_2\}$, $d_2(q_0) = \{\alpha_1, \alpha_2\}$ and one resource $R$ is shown. Action $\alpha_1$ costs one unit of $R$ and action $\alpha_2$ is cost-free; i.e. $t(\alpha_1, R) = -1$ and $t(\alpha_2, R) = 0$.

**Definition 8.38 ($Q^{\leq\omega}$, (resource-extended) path, $\lambda|_Q$, $\lambda|_{\mathcal{R}es}$).** *We define* $Q^{\leq\omega} := Q^\omega \cup Q^+$ *(i.e. all infinite and finite sequences over* $Q$*). A* path $\lambda \in Q^{\leq\omega}$ *is a finite or infinite sequence of states such that there is a transition between two adjacent states.*

*We define a* resource extended path $\lambda$ *as a finite or infinite sequence over* $Q \times \mathsf{En}$ *such that the restriction of* $\lambda$ *to states (the first component), denoted by* $\lambda|_Q$*, is a path in the underlying model. Similarly, we use* $\lambda|_{\mathcal{R}es}$ *to refer to the projection of* $\lambda$ *to the second component of each element in the sequence.*

We would like to note that we do also allow *finite* paths! Intuitively, not all paths are possible given limited resources.

We also use the following notations already introduced for the "standard" notion of paths.

**Definition 8.39 (Length, subpath).** *The* length *of* $\lambda$ *(where* $\lambda$ *is a path or resource extended path), denoted* $l(\lambda)$*, is the number of states in the sequence; if* $\lambda \in Q^\omega$ *then* $l(\lambda) = \infty$*. For* $i \in \mathbb{N}_0$*, we define* $\lambda[i]$ *to be the* $(i+1)$*-th sate* *on* $\lambda$ *or the last one if* $i \geq l(\lambda)$*. Moreover,* $\lambda[i,\infty]$ *refers to the infinite subpath* $\lambda[i]\lambda[i{+}1]\dots$ *of* $\lambda$ *if* $l(\lambda) = \infty$*; or to the finite subpath* $\lambda[i]\lambda[i{+}1]\dots\lambda[l(\lambda){-}1]$ *if* $l(\lambda) < \infty$*. The set of all paths in* $\mathfrak{M}$ *starting in a state* $q$ *is defined as* $\Lambda_{\mathfrak{M}}(q)$*.*

Ultimately, we would like to analyse the ability of groups of agents. We are interested in the existence of a *winning strategy* for a group of agents. As before a *strategy* is a function that fixes the behaviour of an agent; that is, it determines an action for each 'situation' where we will consider two types of situations. Agents can base their decisions on the current state only; or they can base their decision on the whole history. Strategies are defined as in Definition 2.17; however, we simply use *R-strategy* (resp. r-strategy) to refer to an *IR*-strategy (resp. *Ir*-strategy) since we are only interested in the perfect information case. Moreover, we define both strategies over histories of states.

**Definition 8.40 (R/r-strategy).** *A* perfect recall strategy for agent $a$ *(or* R-strategy*) is a function* $s_a : Q^+ \to Act$ *such that* $s_a(q_1 \dots q_n) \in d_a(q_n)$*. A strategy* $s_a$ *is called* memoryless *(or* r-strategy*) if* $s_A(hq) = s_A(h'q)$ *for all* $h, h' \in Q^*$ *and* $q \in Q$ *(as before strategies can be defined as functions* $Q \to Act$*).*

Actions require or produce certain amounts of resources (modelled by $t$) that have to be present for an action to be executed. Agents in a group $A$ can cooperate and *share* their resources, as well can the opponents $\mathbb{A}\text{gt}\backslash A$. In the following, we formalise such 'shares' $\mathsf{sh}$ with respect to an available endowment $\eta$ for some RQM $\rho$.

**Definition 8.41 ($(A,\eta)$-share for $\rho$).** *Let* $\eta$ *be an endowment and let* $\rho$ *be an* RQM*. An* $(A,\eta)$-share for $\rho$ *is a function* $\mathsf{sh} : A \times \mathcal{R}es \to \mathbb{N}_0$ *such that:*

1. $\forall r \in \mathcal{R}es : \rho(r) > 0 \Rightarrow \sum_{a \in A} \mathsf{sh}(a,r) = \rho(r)$ *(the share equals the demand); and*
2. $\forall a \in A, r \in \mathcal{R}es : \eta_a(r) \geq \mathsf{sh}(a,r)$ *(each agent's share must be available).*

A strategy $s_A$ restricts the possible paths in an RBAM; moreover, considering resource-extended paths, only those in which agents have sufficient resources available in each state are feasible. We use the resource component to keep track of the available resources.

We define which extended paths $\lambda$ are possible under a given endowment $\eta$ and strategy $s_A$ assuming agents $A \cup B$ require resources.

**Definition 8.42 ($(\eta, s_A, B)$-path, $out(q,\eta,s_A,B)$).** *An* $(\eta, s_A, B)$*-path is a* maximal *resource-extended path* $\lambda \in (Q \times \mathsf{En})^{\leq\omega}$ *such that for all* $i = 0,\dots,l(\lambda)-2$ *with* $\lambda[i] := (q_i, \eta^i)$ *there is an action profile* $\boldsymbol{\alpha} \in d(\lambda|_Q[i])$ *such that*

1. $\lambda|_{\mathcal{R}es}[0] \leq \eta$ *(initially at most* $\eta$ *resources are available),*

2. $s_A(\lambda|_Q[0,i]) = \boldsymbol{\alpha}|_A$ *(A's actions in* $\boldsymbol{\alpha}$ *are the ones prescribed by strategy* $s_A$*),*
3. $\lambda|_Q[i+1] = o(\lambda|_Q[i], \boldsymbol{\alpha})$ *(transitions are taken according to the action profile* $\boldsymbol{\alpha}$*),*
4. $\forall\ a \in A \forall r \in \mathcal{R}es : (\eta_a^{i+1}(r) = \eta_a^i(r) + \mathsf{prod}(\boldsymbol{\alpha}|_a, r) - \mathsf{sh}(a,r))$ *where* $\mathsf{sh} : A \times \mathcal{R}es \to \mathbb{N}_0$ *is an* $(A,\eta)$*-share for* $r \mapsto \sum_{a\in A} \mathsf{cons}(\boldsymbol{\alpha}|_a, r)$ *(A's resources change according to some appropriate share),*
5. $\forall b \in B\backslash A\ \forall r \in \mathcal{R}es : (\eta_b^{i+1}(r) = \eta_b^i(r) + \mathsf{prod}(\boldsymbol{\alpha}|_b, r) - \mathsf{sh}(b,r))$ *where* $\mathsf{sh} : B\backslash A \times \mathcal{R}es \to \mathbb{N}_0$ *is an* $(B\backslash A,\eta)$*-share for* $r \mapsto \sum_{b\in B\backslash A} \mathsf{cons}(\boldsymbol{\alpha}|_b, r))$ *(*$B\backslash A$*'s resources change according to some appropriate share),*
6. $\forall a \in \mathbb{A}\mathrm{gt}\backslash(A \cup B)\ \forall r \in \mathcal{R}es : (\eta_a^{i+1}(r) = \eta_a^i(r))$ *(available resources remain unchanged for all agents not in* $A \cup B$*),*
7. $\forall a \in \mathbb{A}\mathrm{gt} : ((\lambda|_{\mathcal{R}es}[i])_a \geq 0 \Rightarrow (\lambda|_{\mathcal{R}es}[i+1])_a \geq 0)$ *and* $((\lambda|_{\mathcal{R}es}[i])_a < 0 \Rightarrow (\lambda|_{\mathcal{R}es}[i+1])_a \geq (\lambda|_{\mathcal{R}es}[i])_a)$ *(for each step the required resources are available).*

*We also require condition 1. if* $l(\lambda) = 1$*. The* $\eta$-outcome *of a strategy* $s_A$ *against* $B$ *in* $q$*,* $out(q,\eta,s_A,B)$*, is defined as the set of all* $(\eta,s_A,B)$*-paths starting in* $q$*.*

*Remark 8.43.* (1) We require that a path is maximal, i.e., if a given path can be extended (this is the case if sufficient resources are available) then it must be extended. (2) After an action has been executed the production of resources is added to the endowment of the *action-executing agent.* (3) There are several $(\eta, s_A, B)$-paths due to the choices of the opponents *and* due to different shares in items 4 and 5.

**Proposition 8.44.** *The outcome* $out(q,\eta,s_A,B)$ *is never empty.*

*Proof.* Suppose the outcome is empty. Consider the resource-extended path $\lambda = (q,\eta)$. Due to maximality and emptiness of the outcome there is no move vector that can be executed from $q$ given the resources $\eta$. But then, $\lambda$ is maximal, satisfies condition 1. and trivially all the other conditions. Hence, it would be in the outcome. Contradiction! ■

Finally, we define four semantics for $\mathcal{L}_{RAL^*}$ over triples consisting of an RBAM together with a state and a given endowment for the agents.

**Definition 8.45 ($\models_R$, $\models_r$, $\models_R^\infty$, $\models_r^\infty$, $\mathsf{RAL}_R^*$, $\mathsf{RAL}_r^*$).** *Consider an* RBAM $\mathfrak{M}$*, a state* $q \in Q_{\mathfrak{M}}$*, and an endowment* $\eta$*. The* $R$-semantics *is given by the satisfaction relation* $\models_R$ *defined as follows.*

$\mathfrak{M}, q, \eta \models_R \mathsf{p}$ *iff* $\mathsf{p} \in \pi(q)$*,*
$\mathfrak{M}, q, \eta \models_R \neg\varphi$ *iff* $\mathfrak{M}, q, \eta \not\models_R \varphi$*,*
$\mathfrak{M}, q, \eta \models_R \varphi \wedge \psi$ *iff* $\mathfrak{M}, q, \eta \models_R \varphi$ *and* $\mathfrak{M}, q, \eta \models_R \psi$*,*

$\mathfrak{M}, q, \eta \models_R \langle\langle A\rangle\rangle_C \gamma$ *iff there is an R-strategy* $s_A$ *for* $A$ *such that* $\mathfrak{M}, \lambda, \eta \models_R \gamma$ *for all* $\lambda \in out(q, \eta, s_A, C)$,
$\mathfrak{M}, q, \eta \models_R \langle\langle A\rangle\rangle_C^\zeta \gamma$ *iff there is an R-strategy* $s_A$ *for* $A$ *such that* $\mathfrak{M}, \lambda, \zeta \models_R \gamma$ *for all* $\lambda \in out(q, \zeta, s_A, C)$,
$\mathfrak{M}, \lambda, \eta \models_R \varphi$ *iff* $\mathfrak{M}, \lambda[0], \eta \models_R \varphi$,

*and for path formulae*

$\mathfrak{M}, \lambda, \eta \models_R \neg\gamma$ *iff not* $\mathfrak{M}, \lambda, \eta \models_R \gamma$,
$\mathfrak{M}, \lambda, \eta \models_R \gamma \wedge \chi$ *iff* $\mathfrak{M}, \lambda, \eta \models_R \gamma$ *and* $\mathfrak{M}, \lambda, \eta \models_R \chi$,
$\mathfrak{M}, \lambda, \eta \models_R \bigcirc\gamma$ *iff* $\mathfrak{M}, \lambda[1, \infty], \lambda|_{\mathcal{R}es}[1] \models_R \gamma$ *and* $l(\lambda) > 1$,
$\mathfrak{M}, \lambda, \eta \models_R \gamma \mathcal{U} \chi$ *iff there is* $i \leq l(\lambda)$ *such that* $\mathfrak{M}, \lambda[i, \infty], \lambda|_{\mathcal{R}es}[i] \models_R \chi$ *and for all* $j$ *with* $0 \leq j < i$ *we have* $\mathfrak{M}, \lambda[j, \infty], \lambda|_{\mathcal{R}es}[j] \models_R \gamma$.

*The* r-semantics *(memoryless semantics)* $\models_r$ *is defined similarly to the R-semantics but* r-strategies *are used instead of R-strategies. Moreover, we introduce a variant that focuses on infinite paths. Therefore, in the semantic clauses of the cooperation modalities, we replace "*$\lambda \in out(q, \eta, s_A, C)$*" with "infinite* $\lambda \in out(q, \eta, s_A, C)$*". The resulting semantic relations are denoted* $\models_R^\infty$ *and* $\models_r^\infty$*.*

*The logic* $\mathbf{RAL}_R^*$ *(resp.* $\mathbf{RAL}_r^*$*) is defined as the language* $\mathcal{L}_{RAL^*}$ *together with R-semantics* $\models_R$ *(resp. r-semantics* $\models_r$*).*

The 'infinite semantics' is needed for some extended expressiveness and complexity results. The language $\mathcal{L}_{RAL^*}$, however, is sufficiently expressive to describe infinite paths by "$\Box \bigcirc \top \rightarrow \ldots$" (cf. Proposition 8.54).

*Example 8.46.* Recall the RBAM from Example 8.37 and consider the following endowment $\eta$: $\eta(1)(R) = 2$ and $\eta(2)(R) = \infty$. Then, we have $\mathfrak{M}, q_0, \eta \not\models_r \langle\langle 1\rangle\rangle \Diamond \mathsf{p}$ and $\mathfrak{M}, q_0, \eta \models_r \langle\langle 2\rangle\rangle \Diamond \mathsf{p}$; there are two paths $\lambda$ and $\lambda'$ in the outcome: $\lambda|_Q = q_0(q_2)^\omega$ and $\lambda'|_Q = q_0 q_1 q_1$. But note, that we have $\mathfrak{M}, q_0, \eta \models_r^\infty \langle\langle 1\rangle\rangle \Diamond \mathsf{p}$ as the finite path $\lambda'$ is disregarded.

### 8.4.3 Syntactically Restricted Variants.

Following [Clarke and Emerson, 1981; Alur et al., 2002], we define (temporal) restrictions of $\mathcal{L}_{RAL^*}$.

**Definition 8.47 (Languages $\mathcal{L}_{RAL^+}$ and $\mathcal{L}_{RAL}$).** *The language* $\mathcal{L}_{RAL^+}(\Pi, \mathbb{A}\mathrm{gt}, \mathsf{En})$ *restricts* $\mathcal{L}_{RAL^*}(\Pi, \mathbb{A}\mathrm{gt}, \mathsf{En})$ *in such a way that path formulae are given by* $\gamma ::= \neg\gamma \mid \gamma \wedge \gamma \mid \varphi \mathcal{U} \varphi \mid \bigcirc\varphi$.
*The language* $\mathcal{L}_{RAL}(\Pi, \mathbb{A}\mathrm{gt}, \mathsf{En})$ *is given by*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle_B \bigcirc \varphi \mid \langle\langle A\rangle\rangle_B \Box\varphi \mid \langle\langle A\rangle\rangle_B \varphi \mathcal{U} \varphi \mid \langle\langle A\rangle\rangle_B \varphi \mathcal{R} \varphi \mid \\ \langle\langle A\rangle\rangle_B^\eta \bigcirc \varphi \mid \langle\langle A\rangle\rangle_B^\eta \Box\varphi \mid \langle\langle A\rangle\rangle_B^\eta \varphi \mathcal{U} \varphi \mid \langle\langle A\rangle\rangle_B^\eta \varphi \mathcal{R} \varphi.$$

For the semantic interpretation we consider the 'release' operator as the following macro: $\varphi \mathcal{R} \psi \equiv \neg((\neg\psi)\mathcal{U}(\neg\varphi))$. Differently, to $\mathcal{L}_{CTL}$ and $\mathcal{L}_{ATL}$ we do also allow the 'release' operator $\mathcal{R}$ in $\mathcal{L}_{RAL}$. Note that $\mathcal{L}_{RAL}$ with the release operator is strictly more expressive than without it [Laroussinie et al., 2008]. Let $\mathcal{L}_{RAL}{}'$ denote the sublanguage of $\mathcal{L}_{RAL}$ without the release operator. Then, we have the following result which is obvious from [Laroussinie et al., 2008].

**Proposition 8.48.** *There is no formula $\varphi \in \mathcal{L}_{RAL}{}'$ such that $\varphi \leftrightarrow \langle\langle A\rangle\rangle^{\eta_\infty} \mathsf{r} \mathcal{R} \mathsf{s}$ is valid where $\eta_\infty$ maps every agent and resource type to $\infty$.*

In the following we define variants of all languages that restrict the use of resources. Operators $\langle\langle A\rangle\rangle_B$ assume that the proponents $A$ *and* opponents $B\backslash A$ act under limited resources whereas $\langle\langle A\rangle\rangle$ only restricts the choices of the proponents $A$. In Section 11.3 we show that this affects the model checking proofs.

Another aspect of complexity is reflected by the two cooperation modalities $\langle\langle A\rangle\rangle_C$ and $\langle\langle A\rangle\rangle_C^\eta$. The former operator is intuitively harder to handle than the latter as one has to keep track of resources. We note that the expressiveness of the logic justifies operators of the first kind. For example, consider the formula $\langle\langle A\rangle\rangle\Diamond(\mathsf{p} \wedge \langle\langle B\rangle\rangle\gamma)$: Agents $A$ have to reach a state in which $\mathsf{p}$ holds and in which $B$ can ensure $\gamma$ with the then *remaining resources* for agents $A \cap B$.

Both restrictions have interesting effects on the model checking complexity wrt the number of agents needed to show undecidability.

**Definition 8.49 (Proponent restrictedness; resource flatness).** *Let $\mathcal{L}$ be any of the languages introduced above.*

*(a) The language pr-$\mathcal{L}$,* proponent-restricted *$\mathcal{L}$, is the sublanguage of $\mathcal{L}$ allowing only operators $\langle\langle A\rangle\rangle$ and $\langle\langle A\rangle\rangle^\eta$.*
*(b) The language rf-$\mathcal{L}$,* resource-flat *$\mathcal{L}$, is the sublanguage obtained from $\mathcal{L}$ if only cooperation modalities $\langle\langle A\rangle\rangle_B^\eta$ are allowed (and not $\langle\langle A\rangle\rangle_B$).*

*Analogously to Definition 8.45, we define the logics $\mathsf{RAL}_R$, $\mathsf{RAL}_r$, $\mathsf{RAL}_r^+$, and $\mathsf{RAL}_R^+$ and their proponent-restricted and/or resource-flat variants.*

### 8.4.4 Restricted RBAMs.

In Section 11.3.1 we show that the model checking problem is often undecidable over general RBAMs. Exceptions are the bounded settings presented in the following.

**Definition 8.50 ($k$-bounded for $\eta$, bounded).** *For $k \in \mathbb{N}$, an* RBAM *$\mathfrak{M}$ is said to be* $k$-bounded for endowment *$\eta$ if for each element $(q, \zeta)$ on any $(\eta, s_A, B)$-path for any strategy $s_A$ and $B \subseteq \mathbb{A}\mathrm{gt}$ either $\zeta(a)(r) \leq k$ or*

$\zeta(a)(r) = \infty$ *holds for all resources* $r \in \mathcal{R}es_{\mathfrak{M}}$ *and agents* $a \in \mathbb{A}\text{gt}$. *An* RBAM *is called* bounded for $\eta$ *if it is* $k$*-bounded for* $\eta$ *for some* $k \in \mathbb{N}$.

At first glance such models may seem quite artificial but in fact there are several natural settings resulting in bounded models. We call a model *production-free* if actions can only consume and not produce resources. Clearly, every production-free model is bounded.

There is another way to *enforce* a bounded setting. The definition above is purely structural and obviously not every RBAM is bounded. However, often agents themselves have limited capabilities such that it does not necessarily make sense to allow them to carry arbitrary amounts of resources. Depending on the resource type only a limited number of units may be permitted in any endowment. In this setting one *imposes* the requirement of boundedness to the semantics and simply discards any resources that exceed a given bound. The latter is a *semantic restriction* and has to be inserted into the definition of paths.

**Definition 8.51 ($k$-bounded $(\eta, s_A, B)$-path).** *We define a* $k$-bounded $(\eta, s_A, B)$-path *as in Definition 8.42 but we set*

$$\lambda|_{\mathcal{R}es}[0] \leq \eta_a^0(r) := \min\{k, \eta_a^i(r)\}$$

*and replace conditions 4 and 5 by the following:*

$$\eta_a^{i+1}(r) = \min\{k, \eta_a^i(r) + \mathsf{prod}(\boldsymbol{\alpha}|_a, r) - \ \mathsf{sh}(a, r))\}.$$

*The* $k$-bounded $\eta$-outcome *of a strategy* $s_A$ *in* $q$ *with respect to* $B$, $out_k(q, \eta, s_A, B)$, *is defined as the set of all* $k$*-bounded* $(\eta, s_A, B)$*-paths starting in* $q$.

Finally, we define the *k-bounded* R-semantics $\models_R^k$ (resp. r-semantics $\models_r^k$) as in Definition 8.45 but replace the outcome by the $k$-outcome.

## 8.5 Properties and Expressiveness

For $\mathcal{L}_{ATL}$ (the plain strategic case without resources) it is well-known that if agents have a perfect recall winning strategy they also have a *memoryless* winning strategy (cf. Theorem 2.24). The temporal dimension is sufficiently restricted such that the truth of each formula does only depend on the initial non-looping segment of a path. The next result shows that this is not the case in the presence of resources. The reason for this is that agents may need to perform an action several times until sufficient resources are produced.

**Proposition 8.52.** *There is an* RBAM $\mathfrak{M}$, $q \in Q_{\mathfrak{M}}$, $\eta \in \mathsf{En}$, *and* $\varphi \in \mathcal{L}_{RAL}$ *such that:* $\mathfrak{M}, \eta, q \models_R \varphi$ *and* $\mathfrak{M}, \eta, q \not\models_r \varphi$

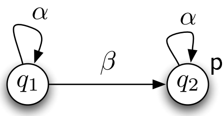



**Fig. 8.5.** Model used in the proof of Proposition 8.52.

*Proof.* Consider a simple model $\mathfrak{M}$ shown in Figure 8.5 with two states $q_1$ and $q_2$ where $\mathsf{p}$ holds in state $q_2$. In order to reach state $q_2$ the agent has to perform an action $\alpha$ resulting in a loop in $q_1$ that produces one unit of a resource needed to execute action $\beta$ that leads to $q_2$. However, such a strategy cannot be achieved with a memoryless strategy, as the agent has to execute first $\alpha$ and then $\beta$ in the very same state $q_1$. Hence, we have $\mathfrak{M}, q_1, \eta_0 \models_R \langle\langle 1\rangle\rangle \Diamond \mathsf{p}$ but $\mathfrak{M}, q_1, \eta_0 \not\models_r \langle\langle 1\rangle\rangle \Diamond \mathsf{p}$. ■

Clearly, due to the semantic definition we have that $\models_R \langle\langle A\rangle\rangle_C \gamma \leftrightarrow \langle\langle A\rangle\rangle_{A\cup C}\gamma$ for any $A, C \subseteq \mathbb{A}\mathrm{gt}$ and $\gamma \in \mathcal{L}_{RAL^*}$. The same holds for $\langle\langle A\rangle\rangle^{\eta}_{C}$.

One easily observes that if actions are cost-free, then each path in the outcome is infinite (due to maximality) and both types of cooperation modalities coincide. Therefore, we obtain the following result.

**Proposition 8.53.** *$\mathcal{L}_{RAL^*}$ (resp. $\mathcal{L}_{RAL}$, $\mathcal{L}_{RAL^+}$) subsumes $\mathcal{L}_{ATL^*}$ (resp. $\mathcal{L}_{ATL}$, $\mathcal{L}_{ATL^+}$) over the R-semantics and r-semantics, respectively.*

As mentioned earlier, the language $\mathcal{L}_{RAL^*}$ is sufficiently expressive to describe infinite paths by "$\Box \bigcirc \top \rightarrow \ldots$". Hence, we can state as a fact that the semantics focusing on infinite paths can be simulated by $\mathcal{L}_{RAL^*}$.

**Proposition 8.54.** *Logic $(\mathcal{L}_{RAL^*}, \models_x)$ subsumes $(\mathcal{L}_{RAL^*}, \models_x^{\infty})$ for $x \in \{r, R\}$. This also holds for the* proponent restrictive *(pr) and* resource flat *(rf) variants of Definition 8.49.*

*Proof.* First, we show that $\mathfrak{M}, \lambda, \eta \models_x \Box \bigcirc \top$ iff $\lambda$ is infinite. Assume $\mathfrak{M}, \lambda, \eta \models_x \Box\bigcirc\top$ and $l(\lambda) = n < \omega$. Then; $\mathfrak{M}, \lambda[n-1, \infty], \eta \not\models \bigcirc\top$. Now, let $\lambda$ be infinite. Since each $\pi(\lambda[i])$ is consistent, we have that $\mathfrak{M}, \lambda[i, \infty], \eta \models_x \bigcirc\top$ for all $i \in \mathbb{N}_0$; hence, $\mathfrak{M}, \lambda, \eta \models_x \Box \bigcirc \top$.

Replace each "$\langle\langle A\rangle\rangle^{\eta}_{B}\gamma$" by "$\langle\langle A\rangle\rangle^{\eta}_{B}(\Box \bigcirc \top \rightarrow \gamma)$" and analogously for $\langle\langle A\rangle\rangle_B$. Then, we have the following:

$$
\begin{aligned}
\mathfrak{M}, q, \eta \models_x^{\infty} \langle\langle A\rangle\rangle^{\eta}_{B}\gamma \text{ iff } & \text{there is an x-strategy } s_A \text{ for } A \text{ s.t. } \mathfrak{M}, \lambda, \eta \models_x^{\infty} \gamma \\
& \quad \text{for all infinite } \lambda \in out(q, \eta, s_A, B) \\
\text{iff } & \text{there is an x-strategy } s_A \text{ for } A \text{ s.t.} \\
& \quad \text{for all } \lambda \in out(q, \eta, s_A, B)\ \mathfrak{M}, \lambda, \eta \models_x \Box \bigcirc \top \rightarrow \gamma \\
\text{iff } & \mathfrak{M}, q, \eta \models_x \langle\langle A\rangle\rangle^{\eta}_{B}(\Box \bigcirc \top \rightarrow \gamma)
\end{aligned}
$$

■

### 8.5.1 Single-Agent Logics

In this section we show that the resource-bounded tree logics introduced above and the logic from [Alechina et al., 2009b], which also deal with resources but in a single agent setting, can be embedded in the resource agent logics presented here. The logic **RTL**$^*$ can be seen as the single agent version of **RAL**$^*$. The operator $\langle\rho\rangle\gamma$ ("there is an infinite path feasible with resources $\rho$") is translated to $\langle\langle \mathbb{A}\mathrm{gt}\rangle\rangle^{\eta_\rho}(\Box\bigcirc\top\wedge\gamma)$. We define *rf*-**RAL**$_{R\infty}$ as *rf*-$(\mathcal{L}_{RAL}, \models^\infty_R)$. Where $\eta_\rho$ equips some agent with the resources in $\rho$. Similarly, *rf*-**RAL**$_{r\infty}$ is defined as *rf*-$(\mathcal{L}_{RAL}, \models^\infty_r)$ and likewise for the proponent-restrictive logics.

For the resource-bounded tree logics resources are modelled by a resource-quantity mapping $\rho$. Models are extensions of Kripke structures. These models are slightly different to the single agent setting of RBAMs, as they allow to consume and produce from a resource in a single step. However, such models can also be modified such that transitions are split into two. The formulae have to be translated as well. Here we give the semantics of the resource-bounded tree logics directly over RBAMs.

**Definition 8.55 ($\rho$-feasible path wrt.** RBAMs**).** *A path $\lambda$ is called $\rho$-*feasible *if there is a strategy $s_{\mathbb{A}\mathrm{gt}}$ of the grand coalition* $\mathbb{A}$gt *such that $\lambda$ is the (unique (with respect to the projection to $Q$)) path $\lambda \in out(q, s_{\mathbb{A}\mathrm{gt}}, \eta^\rho, \mathbb{A}\mathrm{gt})$ and if $\lambda$ is infinite.*

So, the clause for the path quantifier is given as follows:

$$\mathfrak{M}, q, \eta \models \langle\rho\rangle\gamma \text{ iff there is a } \rho\text{-feasible path starting in } q \text{ such that } \mathfrak{M}, \lambda, \eta^\rho \models \gamma.$$

Next, we can show that the resource-bounded tree logics can be embedded in the resource agent logics. The main difference between both logics are the path quantifiers. In the resource-bounded tree logics, the operator $\langle\rho\rangle\gamma$ says that there is an infinite $\rho$-feasible path along which $\gamma$ holds. Hence, we have to characterise infinite paths, this is either possible with the infinity semantics or with $\mathcal{L}_{RAL^*}$.

**Theorem 8.56.** *The single agent fragments of rf-pr-***RAL**$^*_R$ *and rf-***RAL**$^*_r$ *(resp. rf-***RAL**$_{r\infty}$*) embed* **RTL**$^*$ *(resp.* **RTL***) over* RBAMs.

*Proof.* In the following we consider all four cases and show how $\langle\rho\rangle$ can be expressed in the resource agent logics. For this purpose, we define a translation function $tr(\cdot)$ mapping formulae of $\mathcal{L}_{RTL^*}$ to $\mathcal{L}_{RAL^*}$. The cases for propositions, negation, conjunction, etc. are as usual and are not repeated here.

Given a *RQM* $\rho$ of **RTL**$^*$, we define $\eta_\rho$ as the resource endowment (for **RAL**$^*$) such that for some agent $a \in \mathbb{A}\mathrm{gt}$ we have that $\eta_\rho(a, r) = \rho(r)$ and for all other agents $b \in \mathbb{A}\mathrm{gt}\backslash\{a\}$, $\eta_\rho(b, r) = 0$ for all resource types $r \in \mathcal{R}es$. That is, we equip one agent (viz. $a$) with the resources given by $\rho$. The agent can transfer the resources to other agents in the coalition choosing an appropriate share. In the following we concretise $tr(\cdot)$ for the appropriate languages.

*rf-pr-*$\mathbf{RAL}^*_R$ embeds $\mathbf{RTL}^*$. We set $tr(\langle\rho\rangle\gamma) = \langle\langle\mathbb{A}\text{gt}\rangle\rangle^{\eta_\rho}(\Box\bigcirc\top \wedge tr(\gamma))$ and can show the following embedding:

$$\begin{aligned} &\mathfrak{M}, q, \eta \models_R \langle\langle\mathbb{A}\text{gt}\rangle\rangle^{\eta_\rho}(\Box\bigcirc\top \wedge tr(\gamma)) \\ \Leftrightarrow\quad & \exists s_{\mathbb{A}\text{gt}} \forall\lambda \in out(q, \eta_\rho, s_{\mathbb{A}\text{gt}}, \mathbb{A}\text{gt}) : \ \mathfrak{M}, \lambda, \eta \models_R \Box\bigcirc\top \wedge tr(\gamma) \\ \Leftrightarrow\quad & \exists s_{\mathbb{A}\text{gt}} \exists\lambda \in out(q, \eta_\rho, s_{\mathbb{A}\text{gt}}, \mathbb{A}\text{gt}) : \ \mathfrak{M}, \lambda, \eta \models_R \Box\bigcirc\top \wedge tr(\gamma) \\ \Leftrightarrow\quad & \exists s_{\mathbb{A}\text{gt}} \exists \textit{ infinite } \lambda \in out(q, \eta_\rho, s_{\mathbb{A}\text{gt}}, \mathbb{A}\text{gt}) : \ \mathfrak{M}, \lambda, \eta \models_R tr(\gamma) \\ \Leftrightarrow\quad & \exists\rho\textit{-feasible } q\textit{-path } \lambda\textit{:}\ \mathfrak{M}, \lambda, \eta^\rho \models_R tr(\gamma) \\ \Leftrightarrow\quad & \mathfrak{M}, q, \eta \models_R \langle\rho\rangle\gamma. \end{aligned}$$

*rf-*$\mathbf{RAL}^*_r$ embeds $\mathbf{RTL}^*$. We set $tr(\langle\rho\rangle\gamma) = \neg\langle\langle\emptyset\rangle\rangle^{\eta_\rho}_{\mathbb{A}\text{gt}}\neg(\Box\bigcirc\top \wedge tr(\gamma))$. Then, we have that $\mathfrak{M}, q, \eta \models_r \neg\langle\langle\emptyset\rangle\rangle^{\eta_\rho}_{\mathbb{A}\text{gt}}\neg(\Box\bigcirc\top\wedge tr(\gamma))$ iff $\exists\lambda \in out(q, s_\emptyset, \eta^\rho, \mathbb{A}\text{gt})$ such that $\mathfrak{M}, \lambda, \eta^\rho \models_r \Box\bigcirc\top \wedge tr(\gamma)$ iff there is a $\rho$-feasible path $\lambda$ such that $\mathfrak{M}, \lambda, \eta^\rho \models_r tr(\gamma)$.

*rf-*$\mathbf{RAL}_{r\infty}$ embeds $\mathbf{RTL}$. We set $tr(\langle\rho\rangle\gamma) = \neg\langle\langle\emptyset\rangle\rangle^{\eta_\rho}_{\mathbb{A}\text{gt}}\neg tr(\gamma)$. We have $\mathfrak{M}, q, \eta \models^\infty_r \neg\langle\langle\emptyset\rangle\rangle^{\eta_\rho}_{\mathbb{A}\text{gt}}\neg tr(\gamma)$ iff there is an infinite path $\lambda \in out(q, s_\emptyset, \eta^\rho, \mathbb{A}\text{gt})$ such that $\mathfrak{M}, \lambda, \eta^\rho \models^\infty_r tr(\gamma)$ iff there is a $\rho$-feasible path $\lambda$ such that $\mathfrak{M}, \lambda, \eta^\rho \models_r tr(\gamma)$.

■

*rf-pr-*$\mathbf{RAL}_{R\infty}$ does not subsume $\mathbf{RTL}$; at least not in an obvious way. The reason is that $\langle\langle\mathbb{A}\text{gt}\rangle\rangle^{\eta_\rho}$ is not expressive enough to enforce the existence of an infinite path. It only universally quantifies over this set; hence, if there are only finite paths every formula will trivially be true.

### 8.5.2 Multi-Agent Logics

**RBCL** [Alechina et al., 2009b] introduces resources to an extension of coalition logic (cf. Section 2.2.4). Actions are not allowed to produce resources. The main operator $[A^b]\varphi$ is read as follows: Coalition $A$ can enforce $\varphi$ in a finite number of steps given the resources $b$; formally,

> $\mathfrak{M}, q \models_{\mathbf{RBCL}} [A^b]\varphi$ for $A \neq \emptyset$ iff there is a strategy ($R$-strategy in our notation) such that for all $\lambda \in out(q, s_A)$ there is an $m > 0$ such that $cost(\lambda[0, m], s_A) \leq b$ and $\mathfrak{M}, \lambda[m] \models_{\mathbf{RBCL}} \varphi$).

Intuitively, *cost* sums up the transition cost of each step. Resources, however, can be combined in various ways (not only additive); hence, we restrict ourselves to a variant of **RBCL** that will only allow to sum up resource costs, denoted by $\mathsf{RBCL}_+$. Then, the operator $[A^b]$ can be encoded as $\langle\langle A\rangle\rangle^{\eta^b}\bigcirc\langle\langle A\rangle\rangle\Diamond$ for $A \neq \emptyset$. The empty coalition is treated as a special case:

$\mathfrak{M}, q \models_{\mathsf{RBCL}} [\emptyset^b]\varphi$ iff for all strategies $s_{\mathbb{A}\mathrm{gt}}$ ($R$-strategy in our notation) and all $\lambda \in out(q, s_{\mathbb{A}\mathrm{gt}})$ and all $m > 0$ such that $cost(\lambda[0,m], s_{\mathbb{A}\mathrm{gt}}) \leq b$ it holds that $\mathfrak{M}, \lambda[m] \models_{\mathsf{RBCL}} \varphi$).

such we can define $[\emptyset^b]$ as $\langle\langle\emptyset\rangle\rangle^{\eta^b}_{\mathbb{A}\mathrm{gt}} \bigcirc \langle\langle\emptyset\rangle\rangle_{\mathbb{A}\mathrm{gt}}\Box$.

**Theorem 8.57.** $\mathsf{RAL}_R$ *subsumes* $\mathsf{RBCL}_+$.

*Proof.* Let $\mathfrak{M}$ be an **RBCL**-model. This model can directly be translated to an *RBAM* $\mathfrak{M}'$. We recursively replace $[A^b]$ with $\langle\langle A\rangle\rangle^{\eta^b} \bigcirc \langle\langle A\rangle\rangle\Diamond$ for $A \neq \emptyset$ and $[\emptyset^b]$ to $\langle\langle\emptyset\rangle\rangle^{\eta^b}_{\mathbb{A}\mathrm{gt}} \bigcirc \langle\langle\emptyset\rangle\rangle_{\mathbb{A}\mathrm{gt}}\Box$. We prove the case for $\mathfrak{M}, q \models_{\mathsf{RBCL}} [A^b]\mathsf{p}$.

$$\begin{aligned}
&\mathfrak{M}, q \models_{\mathsf{RBCL}} [A^b]\mathsf{p}\\
&\text{ iff } \exists s_A \forall \lambda \in out(q, s_A)\exists m > 0 : (cost(\lambda[0,m], s_A) \leq b \wedge \mathfrak{M}, \lambda[m] \models_{\mathsf{RBCL}} \mathsf{p})\\
&\text{ iff } \exists s_A \forall \lambda \in out(q, \eta^b, s_A, A)\exists m > 0 : \lambda[m] \models_{\mathsf{RBCL}} \mathsf{p})\\
&\text{ iff } \exists s_A \forall \lambda \in out(q, \eta^b, s_A, A)(\mathfrak{M}, \lambda[1], \lambda|_{\mathcal{R}es}[1] \models_{\mathsf{RAL}_R} \langle\langle A\rangle\rangle\Diamond\mathsf{p})\\
&\text{ iff } \mathfrak{M}, q, \eta \models_{\mathsf{RAL}_R} \langle\langle A\rangle\rangle^{\eta^b} \bigcirc \langle\langle A\rangle\rangle\Diamond\mathsf{p}.
\end{aligned}$$

Negation, conjunction, and the case for the empty coalition are treated analogously. ■

The logic resource-bounded **ATL** (**RB-ATL**) introduced in [Alechina et al., 2010] is another proposal for an extension of **ATL** with resources. **RB-ATL** has a decidable model checking property due to the fact that it only allows for the consumption of resources and hence making all models bounded by default. There seems to be a similar encoding of **RB-ATL** formulae into **RAL** formulae.

## 8.6 Summary

In this chapter we have introduced resources into $\mathsf{CTL}^\star$, which is arguably among the most important logics in computer science, and $\mathsf{ATL}^*$, which is among the most influential multi-agent logics.

The resource-bounded tree logics $\mathsf{RTL}^\star$ and **RTL** allow to model reactive systems in the presence of resources. We have also considered bounded settings which we show to have a decidable model checking problem in Chapter 11.

We have presented various strategic logics for reasoning about abilities of *multiple* agents under limited resources. The different settings were based on classical restrictions (cf. [Clarke and Emerson, 1981; Alur et al., 2002]) imposed on the underlying temporal language ($\mathcal{L}_{RAL^*}$ vs. $\mathcal{L}_{RAL^+}$ vs. $\mathcal{L}_{RAL}$) and strategic dimension (perfect vs. imperfect recall). Additionally, we have imposed restrictions on the resource dimension by focussing on specific groups acting under limited resources (proponent-restrictiveness) and on the nesting of cooperation operators (resource-flatness).

Moreover, we have shown that the resource-bounded tree logics can be considered as the single agent fragment of the resource agent logics and how the latter relates to other logics dealing with resources.

# Part III

# Complexity of Verifying Rational Agents

# 9

# Verifying Agents with Memory is Harder Than It Seemed

## Contents



In this chapter we analyse the effect of memory on the model checking complexity of $\mathcal{L}_{ATL^+}$. We correct results from the literature and show that the language can be extended without increasing the model checking complexity.

## 9.1 $\mathsf{ATL}^+_{\mathsf{IR}}$, $\mathsf{EATL}^+_{\mathsf{IR}}$ and The Matter of Recall

In an excellent study [Schobbens, 2004], Schobbens claims that model checking $\mathcal{L}_{ATL^+}$ is $\mathbf{\Delta_3^P}$-complete wrt the number of transitions in the model and the length of the formula, for both perfect recall and memoryless semantics. For memoryless agents, we recall that the upper bound can be shown by the following algorithm (cf. Theorem 5.14). Given a formula $\langle\langle A\rangle\rangle\gamma$ with no nested cooperation modalities, we can guess a memoryless strategy of $A$, "trim" the model accordingly, model check the $\mathsf{CTL}^+$ formula $\mathsf{E}\neg\gamma$ in the resulting model, and revert the result. Note that a memoryless strategy can be guessed in polynomially many steps, and the trimming process requires only polynomially many steps too. For nested cooperation modalities, we repeat the procedure recursively (bottom-up). Since model checking of the $\mathsf{CTL}^+$ formula $\mathsf{E}\neg\gamma$ can be done in nondeterministic polynomial time [Laroussinie et al., 2001], we get that the overall procedure runs in time $\left(\mathbf{P}^{\mathbf{NP}}\right)^{\mathbf{NP}} = \mathbf{P}^{\left(\mathbf{NP}^{\mathbf{NP}}\right)} = \mathbf{\Delta}_{\mathbf{3}}^{\mathbf{P}}$ [Schobbens, 2004].

For agents with perfect recall, a similar argument *seems* to be correct. Each formula of $\mathsf{ATL}^{+}_{\mathsf{IR}}$ can be translated to an equivalent formula of $\mathsf{ATL}_{\mathsf{IR}}$ with weak until [Harding et al., 2002], and for **ATL** (also with weak until) it does not make a difference whether the perfect recall or memoryless semantics is used, so memoryless strategies can be used instead. Hence, it is enough to guess a memoryless strategy, to trim the model etc. Unfortunately, this line of reasoning is wrong because the result of the translation (the $\mathsf{ATL}_{\mathsf{IR}}$ formula) may include *exponentially many* cooperation modalities (instead of one in the original $\mathsf{ATL}^{+}_{\mathsf{IR}}$ formula). For example, formula $\langle\!\langle A\rangle\!\rangle(\Diamond \mathsf{p}_1 \wedge \Diamond \mathsf{p}_2)$ is translated to $\langle\!\langle A\rangle\!\rangle\Diamond\big((\mathsf{p}_1 \wedge \langle\!\langle A\rangle\!\rangle\Diamond\mathsf{p}_2) \vee (\mathsf{p}_2 \wedge \langle\!\langle A\rangle\!\rangle\Diamond\mathsf{p}_1)\big)$; for a longer list of achievement goals $(\Diamond\mathsf{p}_1 \wedge \cdots \wedge \Diamond\mathsf{p}_n)$, each permutation must be explicitly enumerated. Thus, we may need to guess exponentially many polynomial-size strategies, which clearly cannot be done in polynomial time.

There seems to be an intuitive way of recovering from the problem. We note that in an actual execution, only a polynomial number of these strategies will be used. So, we can try to first guess a sequence of goals (in the right order) for which strategies will be needed, then the strategies themselves, fix those strategies in the model (cloning the model into as many copies as we need) and check the corresponding $\mathsf{CTL}^{+}$ formula in it. Unfortunately, this is also wrong: For different execution paths, we may need *different* ordering of the goals (and hence strategies). And we have to consider exponentially many paths in the worst case.

So, what is the complexity of model checking $\mathsf{ATL}^{+}_{\mathsf{IR}}$ in the end? The problem turns out to be (apparently) harder than $\mathbf{\Delta}_{\mathbf{3}}^{\mathbf{P}}$, namely **PSPACE**-complete (unless the polynomial hierarchy collapses).

## 9.2 Model Checking $\mathsf{ATL}^{+}_{\mathsf{IR}}$

### 9.2.1 Lower Bound

We prove the **PSPACE**-hardness by a reduction of Quantified Boolean Satisfiability (QSAT), a canonical **PSPACE**-complete problem (cf. Definition 4.13). We assume that a QSAT instance is given in negation normal form (i.e., negations occur only at literals).

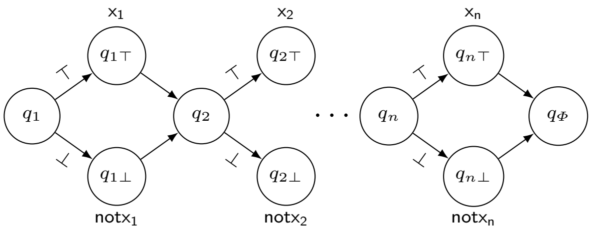



**Fig. 9.1.** Construction of the concurrent game structure for QSAT: value choice section.

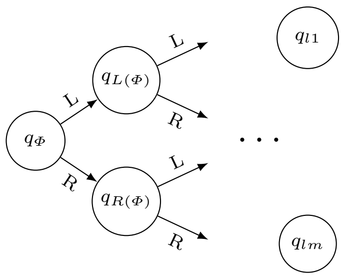



**Fig. 9.2.** CGS for QSAT: formula structure section.

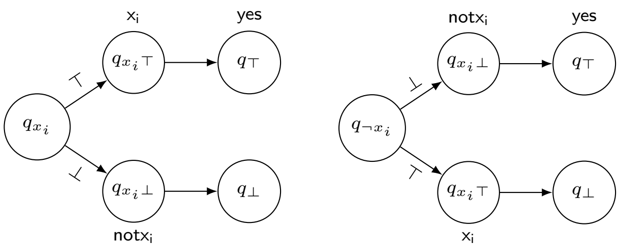



**Fig. 9.3.** CGS for QSAT: sections of literals.

Given an instance of QSAT we construct a turn-based[1] concurrent game structure $\mathfrak{M}$ with two players: the *verifier* **v** and the *refuter* **r**. The structure consists of the following sections:

- *Value choice section:* A sequence of states $q_i$, one per variable $x_i$, where the values of $x_i$'s will be "declared", see Figure 9.1. States $q_i$ with odd $i$ are controlled by **v**, states with even $i$ are controlled by **r**. The owner of a

[1] A model is *turn-based* if for each state there is a single agent that controls the subsequent transition, and the other agents have no real choice there (which can be modelled by assuming $d_q(a) = \{wait\}$ for every agent $a$ except the "owner" of $q$).

state can choose between two possible valuations ($\top, \bot$). Choosing $\top$ leads to a state where proposition $\mathsf{x_i}$ holds; choosing $\bot$ leads to a state labeled by proposition $\mathsf{notx_i}$.

- *Formula structure section:* Corresponds to the parse tree of $\Phi$, see Figure 9.2. For each subformula $\Psi$ of $\Phi$, there is a state $q_\Psi$ with two choices: $L$ leading to state $q_{L(\Psi)}$ and $R$ leading to $q_{R(\Psi)}$, where $L(\Psi)$ is the left hand side subformula of $\Psi$ and $R(\Psi)$ is the right hand side subformula of $\Psi$. The verifier controls $q_\Psi$ if the outermost connective in $\Psi$ is a disjunction; the refuter controls the state if it is a conjunction. Note that each leaf state in the tree is named according to a literal $l_i$ from $\Phi$, that is, either with a variable $x_i$ or its negation $\neg x_i$.
- *Sections of literals:* For each literal $l$ in $\Phi$, we have a single state $q_l$, controlled by the owner of the Boolean variable $x_i$ in $l$. Like in the value choice section, the agent chooses a value ($\top$ or $\bot$) for the variable (not for the literal!) which leads to a new state labeled with the proposition $\mathsf{x_i}$ (for action $\top$) or $\mathsf{notx_i}$ (for $\bot$). Finally, the system proceeds to the winning state $q_\top$ (labeled with the proposition $\mathsf{yes}$) if the valuation of $x_i$ makes the literal $l$ true, and to the losing state $q_\bot$ otherwise – see Figure 9.3 for details, and Figure 9.4 for an example of the whole construction.

  Note that, if the values of variables $x_i$ are assigned uniformly at states $q_l$ (that is, the actions executed at $q_l$ form a valuation of $x_1, \dots, x_n$), then the formula structure section together with the sections of literals implement the game theoretical semantics [Hintikka, 1973] of formula $\Phi$ given the valuation.

Note that the value of variable $x_i$ can be declared twice during an execution of the model (first in the value choice section, and then in the section of literals). The following "consistency" macro: $\mathsf{Cons}_i \equiv \Box\neg\mathsf{x_i} \vee \Box\neg\mathsf{notx_i}$ expresses that the value of $x_i$ cannot be declared both $\top$ and $\bot$ during a single execution. Now, for the *IR*-semantics, we have that:

**Lemma 9.1.** $\exists x_1 \forall x_2 \dots Q_n x_n\ \Phi$ *iff*

$$M, q_1 \models \langle\langle \mathbf{v} \rangle\rangle \Big( \bigwedge_{i \in Odd} \mathsf{Cons}_i \wedge \big( \bigwedge_{i \in Even} \mathsf{Cons}_i \rightarrow \Diamond \mathsf{yes} \big) \Big).$$

*Proof.* The informal idea is as follows. The $\textbf{ATL}^+$ formula specifies that $\mathbf{v}$ can consistently assign values to "its" variables, so that if $\mathbf{r}$ consistently assigns values to "its" variables (in any way), formula $\Phi$ will always evaluate to $\top$, which is exactly the meaning of QSAT. The way a player assigns a value to variable $x_i$ may depend on what has been assigned to $x_1, \dots, x_{i-1}$. We note that this is the reason why perfect recall is necessary to obtain the reduction.

The statement can be formally proved in the following way. We consider wlog only QSAT instances with even alternations of quantifiers (the case

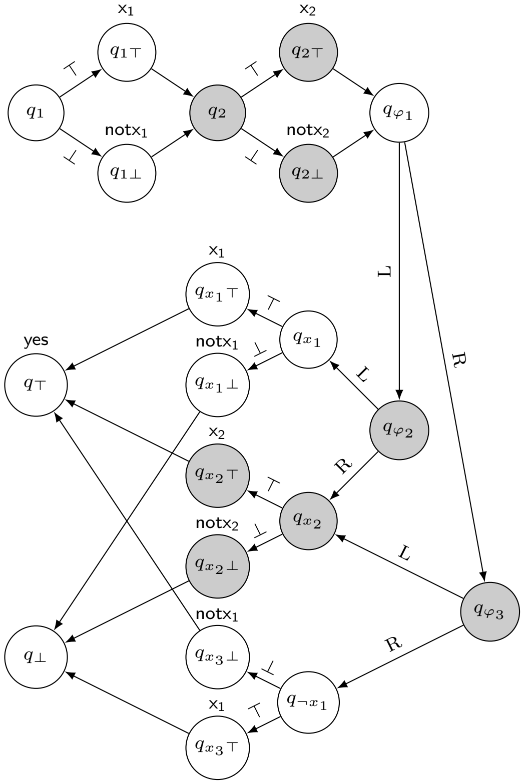



**Fig. 9.4.** Concurrent game structure for the QSAT instance $\exists x_1 \forall x_2 (x_1 \wedge x_2) \vee (x_2 \wedge \neg x_1)$. The following shorthands are used in the formula structure section: $\varphi_1 \equiv (x_1 \wedge x_2) \vee (x_2 \wedge \neg x_1), \varphi_2 \equiv x_1 \wedge x_2, \varphi_3 \equiv x_2 \wedge \neg x_1$. "White" states are owned by the verifier; "grey" states are owned by the refuter.

for odd $n$ is done analogously). Firstly, we note that a QSAT instance $\varphi = \exists x_1 \forall x_2 \ldots \forall x_n \Phi(x_1, \ldots, x_n)$ evaluates to true iff there is a (partial) function $f : \{\top, \bot\}^* \to \{\top, \bot\}$ such that for all valuations $v_i$ of all $x_i$ with $i \leq n$ and $i$ even the following formula is valid:

$$\Phi(f(\epsilon), v_2, f(v_1 v_2), v_3, \ldots, f(v_1 \ldots v_{n-1}), v_n)$$

where $v_1 := f(\epsilon)$, $v_3 := f(v_1 v_2)$,..., $v_{n-1} := f(v_1 \ldots v_{n-2})$. It is easy to see that if such a function exists then it provides a satisfying valuation for the existential quantifiers in the QSAT instance. Conversely, non-existence of such a function contradicts the existence of such a valuation. We say that $f$ *witnesses* $\varphi$. Given a word $w = w_1 \ldots w_m \in \{\top, \bot\}^m$ of length $m \geq 2i$, we use $f_{2i+1}(w)$ to denote the value $f(w_1 \dots w_{2i})$ and $f_{2i}(w)$ to denote $w_{2i}$. Intuitively, $f_i(w)$ returns the assignment $v_i$ of $x_i$ given the choices made before.

"⇒": Let $f$ be a witness for $\varphi$. We define the following strategy $s_{\mathbf{v}}$ for $\mathbf{v}$ for histories $h$ of the form $q_1 q^1 q_2 q^2 \dots$ of length at most $2n$ and each $q^i \in \{q_{i\top}, q_{i\bot}\}$. For states in which only one action is applicable we suppose that it is chosen by default. That is, $h$ is a finite path through the value choice section of length at most $2n$. Moreover, we define mapping $\delta : Q^* \to \{\top, \bot\}^*$ to map each sequence of states to a word over $\{\top, \bot\}$ as follows: $\delta(\epsilon) = \epsilon$, $\delta(q_{i\top}) = \top$, $\delta(q_{i\bot}) = \bot$, and $\delta(q) = \epsilon$ for all other states; finally, $\delta(qh) = \delta(q)\delta(h)$. Then, we define

$$s_{\mathbf{v}}(hq) := \begin{cases} f(\delta(h)) & \text{for } q = q_{2i+1}, \\ nop & \text{for } q \in \{q_{2i+1\top}, q_{2i+1\bot}\}. \end{cases}$$

In each subformula state $q_\psi$ "owned" by $\mathbf{v}$, the verifier chooses action $L$ (resp. $R$) if $L(\psi)$ (resp. $R(\psi)$) evaluates to true given $f$ and $h$ (we write $\psi(f,h) = \top$ for $\psi(f_1(\delta(h)), \dots, f_n(\delta(h)))$ is true):

$$s_{\mathbf{v}}(hq_\psi) := \begin{cases} L & \text{if } L(\psi)(f,h) = \top, \\ R & \text{else.} \end{cases}$$

Analogously, the action for literal states is chosen

$$s_{\mathbf{v}}(hq_l) := \begin{cases} \top & \text{if } l = x_i, f_i(\delta(h)) = \top) \vee \\ & \quad (l = \neg x_i, f_i(\delta(h)) = \bot)), \\ \bot & \text{else.} \end{cases}$$

It remains to show that $s_{\mathbf{v}}$ is a winning strategy of $\mathbf{v}$. Firstly, it is easily seen that $\bigwedge_{i \in Odd} \mathsf{Cons}_i$ holds for any path of the outcome $out(q_1, s_{\mathbf{v}})$; assuming the contrary contradicts the definition of $f$. Finally, assuming that there is a counter strategy of the refuter such that state $q_\bot$ is reached and $\bigwedge_{i \in Even} \mathsf{Cons}_i$ also contradicts that $f$ witnesses $\varphi$ (this can be shown by structural induction on $\Phi$).

"⇐": Let $s_{\mathbf{v}}$ be a winning strategy for $\mathbf{v}$. We define function $f$ in the obvious way, i.e., $f(\epsilon) := s_{\mathbf{v}}(q_1)$, $f(f(\epsilon)\top) := s_{\mathbf{v}}(q_1 q_{(1 f(\epsilon))} q_2 q_{2\top} q_3)$, $f(f(\epsilon)\bot) := s_{\mathbf{v}}(q_1 q_{(1 f(\epsilon))} q_2 q_{2\bot} q_3)$, etc. We prove that $f$ is a witness for $\varphi$.

First, let us observe that for each path in $out(q_1, s_{\mathbf{v}})$, it must hold that $\Box\neg\mathsf{x_i} \vee \Box\neg\mathsf{notx_i}$. As a consequence, $s_{\mathbf{v}}(q_1 \dots q_{x_i}) = s_{\mathbf{v}}(q_1 \dots q_{\neg x_i}) = s_{\mathbf{v}}(q_1 \dots q_i)$ for all histories leading to a literal state $q_l \in \{q_{x_i}, q_{\neg x_i}\}$.[2] That is, the formula structure section and the sections of literals define the game semantics of $\Phi$ with the valuation of $x_1, \dots, x_n$ given uniformly by the $f$ above.

[2] Technically, that is true only for the literal states reachable in $out(q_1, s_{\mathbf{v}})$, but since the unreachable states are irrelevant, we can fix $s_{\mathbf{v}}(q_1 \dots q_l) := s_{\mathbf{v}}(q_1 \dots q_i)$ for unreachable $q_l$'s too.

Moreover, each path in $out(q_1, s_{\mathbf{v}})$ must lead to a state where yes holds (i.e., to $q_\top$), which means that $\Phi$ evaluates to $\top$ given $f$, and thus $f$ is a witness for $\varphi$. ■

We observe that the construction results in a model with $O(|\Phi|)$ states and transitions, and it can be constructed in $O(|\Phi|)$ steps, so we get the following result where the size of a CGS is defined as the number of its transitions ($m$) plus the number of states ($n$). Note, that $\mathcal{O}(n+m) = \mathcal{O}(m)$.

**Theorem 9.2.** *Model checking* $\mathbf{ATL}^+$ *with the perfect recall semantics is* **PSPACE***-hard with respect to the size of the model and the length of the formula. It is* **PSPACE***-hard even for turn-based models with two agents and "flat"* $\mathbf{ATL}^+$ *formulae, i.e., ones that include no nested cooperation modalities.*

### 9.2.2 Upper Bound

In this section we show that model checking $\mathbf{ATL}^+_{\mathbf{IR}}$ can be done in polynomial space. Our proof has been inspired by the construction in [Laroussinie et al., 2001], proposed for $\mathbf{CTL}^+$. We begin by introducing some notation.

We say that $s_A$ is a *strategy for* $(\mathfrak{M}, q, \gamma)$ if for all $\lambda \in out_{\mathfrak{M}}(q, s_A)$ it holds that $\mathfrak{M}, \lambda \models \gamma$. An $\mathbf{ATL}^+$-path formula $\gamma$ is called *atomic* if it has the form $\bigcirc\varphi_1$ or $\varphi_1 \mathcal{U} \varphi_2$ where $\varphi_1, \varphi_2 \in \mathbf{ATL}^+$. For $\varphi \in \mathbf{ATL}^+$ we denote the set of all atomic path subformulae of $\varphi$ by $\mathcal{APF}(\varphi)$. And, as before, we call an $\mathbf{ATL}^+$-path formula $\gamma$ *flat* if it does not contain any more cooperation modalities.

Now we can define the notion of *witness position* which is a specific position on a path that "makes" a path formula true or false.

**Definition 9.3 (Witness position).** *Let $\gamma$ be a flat atomic path formula, and let $\lambda$ be a path. The* witness position $witpos(\lambda, \gamma)$ *of $\gamma$ wrt $\lambda$ is defined as follows:*
*(1) if $\gamma = \bigcirc\varphi$ then $witpos(\lambda, \gamma) = 1$;*
*(2) if $\gamma = \varphi_1 \mathcal{U} \varphi_2$ and*

- $\lambda \models \gamma$ *then* $witpos(\lambda, \gamma) = \min\{i \geq 0 \mid \lambda[i] \models \varphi_2\}$
- $\lambda \not\models \gamma$ *and* $\lambda \models \Diamond\varphi_2$ *then* $witpos(\lambda, \gamma) = \min\{i \geq 0 \mid \lambda[i] \models \neg\varphi_1\}$
- $\lambda \not\models \gamma$ *and* $\lambda \not\models \Diamond\varphi_2$ *then* $witpos(\lambda, \gamma) = -1$.

*Moreover, for a flat (not necessarily atomic)* $\mathbf{ATL}^+$ *path formula $\gamma$, we define the set of witness positions of $\gamma$ wrt $\lambda$ as*

$$wit(\lambda, \gamma) = \Big(\bigcup_{\gamma' \in \mathcal{APF}(\gamma)} \{witpos(\lambda, \gamma')\}\Big) \cap \mathbb{N}_0.$$

For instance, if formula $\Box\neg\mathsf{p}$ is true on $\lambda$ then $\mathit{witpos}(\lambda, \Box\neg\mathsf{p}) = -1$ since the formula is an abbreviation for $\neg(\top\mathcal{U}\mathsf{p})$, and for this formula we have that $\mathit{witpos}(\lambda, \Box\neg\mathsf{p}) = -1$ and consequently, $\mathit{wit}(\lambda, \Box\neg\mathsf{p}) = \emptyset$. In the following we assume that $\gamma$ is flat.

In the next lemma we show that if there is a strategy that enforces a (flat) path formula $\gamma$ then the witnesses of all atomic subformulae of $\gamma$ can be found in a bounded initial fragment of each resulting path. Firstly, we introduce the notion of a *segment* which can be seen as a "minimal loop".

**Definition 9.4 (Segment).** *A* segment *of path $\lambda$ is a tuple $(i, j) \in \mathbb{N}_0^2$ with $i < j$ such that $\lambda[i] = \lambda[j]$ and there are no indices $k, k'$ with $i \leq k < k' \leq j$ such that $\lambda[k] = \lambda[k']$ except for $k = i, k' = j$. The set of segments of $\lambda$ is denoted by $\mathit{seg}(\lambda)$.*

**Lemma 9.5.** *Let $\mathfrak{M}, q \models \langle\!\langle A \rangle\!\rangle\gamma$. Then, there is a strategy $s_A$ for $(\mathfrak{M}, q, \gamma)$ such that for all paths $\lambda \in \mathit{out}_{\mathfrak{M}}(q, s_A)$ the following property holds: For each segment $(i, j) \in \mathit{seg}(\lambda)$ with $j \leq \max \mathit{wit}(\lambda, \gamma)$ there is a witness position $k \in \mathit{wit}(\lambda, \gamma)$ with $i \leq k \leq j$.*

*Proof.* Suppose such a strategy does not exist; then, for any strategy $s_A$ for $(\mathfrak{M}, q, \gamma)$, there is a path $\lambda \in \mathit{out}(q, s_A)$ and a segment $(i, j) \in \mathit{seg}(\lambda)$ with $j \leq \max \mathit{wit}(\lambda, \gamma)$ s.t. there is no $k \in \mathit{wit}(\lambda, \gamma)$ with $i \leq k \leq j$.

We now define $s'_A$ as the strategy that is equal to $s_A$ except that it cuts out the "idle" segment $(i, j)$ from $\lambda$, i.e., $s'_A(\lambda[0, i]h) := s_A(\lambda[0, j]h)$ for all $h \in Q^+$, and $s'_A(h) := s_A(h)$ otherwise. Note that $\mathit{out}(q, s'_A) = \mathit{out}(q, s_A)$ except for paths that begin with $\lambda[0, j]$: These are replaced with paths that achieve the remaining witness positions in $j - i$ less steps. Let $[h]_{q,s_A}$ denote the set of all paths $\lambda'$ such that $h\lambda' \in \mathit{out}(q, s_A)$ where $h \in Q^+$. Now it is easy to see that for all $\lambda' \in [\lambda[0, j]]_{q,s_A}$ we have that the path $\lambda[0, j]\lambda'$ satisfies $\gamma$ if, and only if, the path $\lambda[0, i]\lambda'$ does. Hence, we have that all paths in $\mathit{out}(q, s'_A)$ satisfy $\gamma$. Moreover, the latter set of outcomes is non-empty iff $\mathit{out}(q, s_A)$ is non-empty. By following this procedure recursively, we obtain a strategy that reaches a witness in every segment of each $\lambda$ up to $\max \mathit{wit}(\lambda, \gamma)$. ■

Given, for instance, an $\mathbf{ATL}^+$ formula $\langle\!\langle A \rangle\!\rangle(\Diamond\mathsf{p} \wedge \Diamond\mathsf{r})$ the previous lemma says that if $A$ has any winning strategy than it also has one such that only the first two segments on each path in the outcome are important to witness the truth of $\Diamond\mathsf{p} \wedge \Diamond\mathsf{r}$. In the next definition we make this intuition formal and define the truth of $\mathbf{ATL}^+$ path formulae on finite initial sequences of states.

**Definition 9.6 ($\models^k$).** *Let $\mathfrak{M}$ be a* CGS, *$\lambda$ be path in $\mathfrak{M}$, and $k \in \mathbb{N}$. The semantics $\models^k$ is defined as follows:*

$\mathfrak{M}, \lambda \models^k \neg\gamma$ *iff* $\mathfrak{M}, \lambda \not\models^k \gamma$;
$\mathfrak{M}, \lambda \models^k \gamma \wedge \delta$ *iff* $\mathfrak{M}, \lambda \models^k \gamma$ *and* $\mathfrak{M}, \lambda \models^k \delta$;

$\mathfrak{M}, \lambda \models^k \bigcirc\varphi$ *iff* $\mathfrak{M}, \lambda[1] \models \varphi$ *and* $k > 1$*; and*
$\mathfrak{M}, \lambda \models^k \varphi\mathcal{U}\psi$ *iff there is an* $i < k$ *such that* $\mathfrak{M}, \lambda[i] \models \psi$ *and* $\mathfrak{M}, \lambda[j] \models \varphi$ *for all* $0 \leq j < i$*;*

Essentially, we consider the first $k$ states on a path in order to see whether a formula is made true on it.

We define the notion *$k$-witness positions* on a finite segment of length $k$ in an obvious way: if the witness of $\gamma$ on the full path $\lambda$ is $> k$ then the $k$-witness is $-1$; otherwise, it is equal to $wit(\lambda, \gamma)$.

**Definition 9.7 ($k$-witness strategy).** *We say that a strategy $s_A$ is a $k$-*witness *strategy for $(\mathfrak{M}, q, \gamma)$ if for all $\lambda \in out(q, s_A)$ we have that $\mathfrak{M}, \lambda \models^k \gamma$.*

The following theorem is essential for our model checking algorithm. The result ensures that the existence of a winning strategy can be decided by only guessing the first $k$-steps of a $k$-witness strategy.

**Theorem 9.8.** $\mathfrak{M}, q \models \langle\langle A\rangle\rangle\gamma$ *iff there is a $|Q_{\mathfrak{M}}| \cdot |\mathcal{APF}(\gamma)|$-witness strategy for $(\mathfrak{M}, q, \gamma)$.*

*Proof.* "⇒:" Let $s_A$ be a strategy for $(\mathfrak{M}, q, \gamma)$. By Lemma 9.5 and the fact that $|wit(\lambda, \gamma)| \leq |\mathcal{APF}(\gamma)|$ for any path $\lambda$ there is a strategy $s'_A$ for $(\mathfrak{M}, q, \gamma)$ such that $\max wit(\lambda, \gamma) \leq |Q_{\mathfrak{M}}| \cdot |\mathcal{APF}(\gamma)|$ for all $\lambda \in out(q, s_A)$. This shows that $s'_A$ is a $|Q_{\mathfrak{M}}| \cdot |\mathcal{APF}(\gamma)|$-witness strategy for $(\mathfrak{M}, q, \gamma)$.

"⇐": Suppose there is a $k := |Q_{\mathfrak{M}}| \cdot |\mathcal{APF}(\gamma)|$ witness strategy then there also is a $k$-witness strategy such that on no path in the outcome there is an "idle" segment $(i, j)$ (a segment containing no witness) with $j \leq v$, where $v$ is the maximal witness on the path smaller than $k$ (cf. Lemma 9.5). We call such strategies *efficient*. Now suppose there is an efficient $k$-witness strategy $s_A$ but no strategy for $(M, q, \gamma)$; i.e. for all efficient $k$-witness strategies there is a path $\lambda \in out(q, s_A)$ such that $\mathfrak{M}, \lambda \not\models \gamma$. Note, that this can only happen if there is some $\gamma' \in \mathcal{APF}(\gamma)$ with (minimal) $w := witpos(\lambda, \gamma') \geq k$ that cannot be prevented by $A$ (cf. Figure 9.5). Due to efficiency all subformulae that have a witness $\leq k$ actually have a witness $\leq k - |Q_{\mathfrak{M}}|$. But then, the opponents can ensure that there is some other path $\lambda' \in out(q, s_A)$ on which $\gamma'$ is witnessed within the first $k$ steps on $\lambda'$ and after all the other formulae with a witness $\leq v$ (i.e. within steps $v$ and $k$). This contradicts that $s_A$ is a $k$-witness strategy.

To see that the opponents can ensure that $\gamma'$ is witnessed within the first $k$ steps, consider a segment $(i, j)$ such that $j \leq w$ and $j$ maximal. In particular, the proponents cannot prevent the sequence $\lambda[j, w]$. But then, the opponents are able to execute there moves played from $\lambda[j]$ onwards already from $\lambda[i]$. This results in a path $\lambda''$ also belonging to the outcome which equals $\lambda$ but segment $(i, j)$ being cut out. Following this procedure recursively shows that

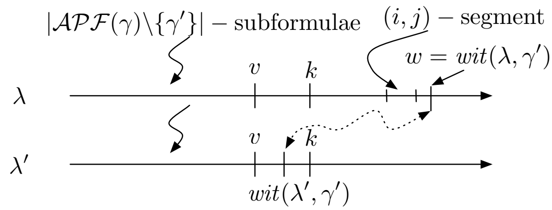



**Fig. 9.5.** Proof of Theorem 9.8.

there is a path $\lambda'$ such that $\gamma'$ is witnessed within the first $k$ steps as stated above.

■

In the next theorem we construct an alternating Turing machine that solves the model checking problem.

**Theorem 9.9.** *Let* $\varphi \equiv \langle\langle A\rangle\rangle\gamma$ *be a flat* $\mathbf{ATL}^{+}_{\mathbf{IR}}$ *formula,* $\mathfrak{M}$ *a* CGS, *and* $q$ *a state. Then, there is a polynomial time alternating Turing machine with* $\mathcal{O}(nl)$ *alternations (wrt the size of the model and length of the formula) that returns "yes" if* $\mathfrak{M}, q \models \varphi$*, and "no" otherwise (where* $l$ *is the length of* $\varphi$*,* $k$ *is the number of agents, and* $n$ *the number of states in* $\mathfrak{M}$*).*

*Proof.* The idea behind the algorithm can be summarised as follows: Coalition $A$ acts as a collective "verifier", and the rest of the agents plays the role of a collective "refuter" of the formula. We first transform $\gamma$ to its negation normal form.[3] Next, we allow the verifier to nondeterministically construct $A$'s strategy step by step for the first $|Q_{\mathfrak{M}}| \cdot |\mathcal{APF}(\gamma)|$ rounds ($|\mathbb{A}\mathrm{gt}|$ steps each), while the refuter guesses the most damaging responses of $\mathbb{A}\mathrm{gt} \setminus A$. This gives us a finite path $h$ (of length $|Q_{\mathfrak{M}}| \cdot |\mathcal{APF}(\gamma)|$) that is the outcome of the best strategy of $A$ against the worst course of events. Then, we implement the game-theoretical semantics of propositional logic [Hintikka, 1973] as a game between the verifier (who controls disjunction) and the refuter (controlling conjunction). The game reduces the truth value of $\gamma$ to a (possibly negated) atomic subformula $\gamma_0$. Finally, we check if $h \models^{|Q_{\mathfrak{M}}|\cdot|\mathcal{APF}(\gamma)|} \gamma_0$, and return the answer. The correctness of the construction follows from Theorem 9.8. ■

For model checking arbitrary $\mathbf{ATL}^{+}_{\mathbf{IR}}$ formulae, we observe that nested cooperation modalities can be model checked recursively (bottom-up) in the same way as e.g. in the standard model checking algorithm for **ATL** [Alur et al., 2002]. Since $\mathbf{P}^{\mathbf{PSPACE}} = \mathbf{PSPACE}$, we obtain the following as immediate corollary.

[3] I.e., so that negation occurs only in front of atomic path subformulae.

**Theorem 9.10.** *Model checking* $\mathbf{ATL}^+_{\mathbf{IR}}$ *over* CGSs *is* **PSPACE**-*complete wrt the size of the model and the length of the formula. It is* **PSPACE**-*complete even for turn-based models with two agents and "flat"* $\mathbf{ATL}^+_{\mathbf{IR}}$ *formulae.*

## 9.3 Correcting Related Results

Concurrent game structures specify transitions through a function that defines state transformations for *each combination of simultaneous actions* from $\mathbb{A}\mathrm{gt}$. In other words, transitions are given through an array that defines the outcome state for each combination of a state with $k$ actions available at that state. This is clearly a disadvantage from the computational point of view, since the array is in general exponential with respect to the number of agents: More precisely, we have that $m = O(nd^k)$, where $m$ is the number of (labeled) transitions in the model, $n$ is the number of states, $d$ is the maximal number of choices per state, and $k$ is the number of agents.

Two variants of game structures overcome this problem. In *alternating transition systems* (ATS), used as models in the initial semantics of **ATL** [Alur et al., 1997, 1998b], agents' choices are state transformations themselves rather than abstract labels. In *implicit concurrent game structures* [Laroussinie et al., 2008], the transition array is defined by Boolean expressions. ATS and implicit CGS do not hide exponential blowup in a parameter of the model checking problem ($m$), and hence the complexity of model checking for these representations is perhaps more meaningful than the results obtained for "standard" CGS. In [Laroussinie et al., 2008], Laroussinie et al. claim that model checking $\mathbf{ATL}^+_{\mathbf{IR}}$ against ATS as well as implicit CGS is $\mathbf{\Delta^P_3}$-complete. Since the proofs are actually based on the flawed result from [Schobbens, 2004], both claims are worth a closer look. We will briefly summarise both kinds of structures and give correct complexity results in this section.

**Alternating Transition Systems.** An ATS is a tuple $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, \delta \rangle$, where $\mathbb{A}\mathrm{gt}, Q, \Pi, \pi$ are like in a CGS, and $\delta : Q \times \mathbb{A}\mathrm{gt} \to \mathcal{P}(\mathcal{P}(Q))$ is a function that maps each pair $(state, agent)$ to a non-empty family of choices with respect to possible next states. The idea is that, at state $q$, agent $a$ chooses a set $Q_a \in \delta(q, a)$ thus forcing the outcome state to be from $Q_a$. The resulting transition leads to a state which is in the intersection of all $Q_a$ for $a \in \mathbb{A}\mathrm{gt}$. Since the system is required to be deterministic (given the state and the agents' decisions), $Q_{a_1} \cap \ldots \cap Q_{a_k}$ must always be a singleton.

**Implicit** CGS**.** We recall from Section 5.4.1, that an implicit CGS is a concurrent game structure where, in *each* state $q$, the outgoing transitions are defined by a finite sequence

$$((\varphi_1, q_1), \ldots, (\varphi_n, q_n)).$$

In the sequence, each $q_i$ is a state, and each $\varphi_i$ is a Boolean combination of propositions $\hat{\alpha}^a$, where $\alpha \in d(a,q)$; $\hat{\alpha}^a$ stands for "agent $a$ chooses action $\alpha$". The transition function is now defined as: $o(q, \alpha_1, ..., \alpha_k) = q_i$ *iff $i$ is the lowest index such that* $\{\hat{\alpha_1}^1, ..., \hat{\alpha_k}^k\} \models \varphi_i$. It is required that $\varphi_n \equiv \top$, so that no deadlock can occur. The *size* of an implicit model is given by the number of states, agents, and the length of the sum of the sizes of the Boolean formulae.

**Model Checking $\mathsf{ATL}^+$ Is PSPACE-Complete Again.** Contrary to [Laroussinie et al., 2008, Section 3.4.1], where model checking $\mathsf{ATL}^+_{\mathsf{IR}}$ with respect to both ATS and implicit CGS is claimed to be $\mathbf{\Delta_3^P}$-complete, we establish the complexity as **PSPACE**.

**Theorem 9.11.** *Model checking* $\mathsf{ATL}^+_{\mathsf{IR}}$ *for* ATS *and implicit* CGS *is* **PSPACE**-*complete wrt the size of the model and the length of the formula (even for turn-based models with two agents and "flat"* $\mathsf{ATL}^+_{\mathsf{IR}}$ *formulae).*

*Proof.* [sketch] *Lower bound.* We observe that the number of transitions in a turn-based CGS is linear in the number of states ($n$), agents ($k$), and actions ($d$). Moreover, each turn-based CGS has an isomorphic ATS, and an isomorphic implicit CGS; the transformation takes $O(nd)$ steps. This, together with the reduction from Section 9.2.1, gives us **PSPACE**-hardness wrt $n, k, d$ and the length of the formula ($l$) and the encoded transition function for model checking $\mathsf{ATL}^+_{\mathsf{IR}}$ against ATS as well as implicit CGS.

*Upper bound.* A close inspection of the algorithm from Section 9.2.2 reveals that it can easily applied to ATS and implicit CGS. In each step when a transition is taken one has to evaluate a sequence of Boolean formulae. This can be done in polynomial time wrt to $a$, $k$, and the length of the encoding.■

## 9.4 Model Checking $\mathsf{EATL}^+_{\mathsf{IR}}$

In this section we extend the construction from Section 9.2.2 to obtain an algorithm for $\mathcal{L}_{EATL^+}$ under the perfect recall semantics. Firstly, we define the *set of witnesses* $wit^\infty(\lambda, \gamma)$ for a flat atomic formula $\gamma \equiv \overset{\infty}{\Diamond}\varphi$. If $\lambda \not\models \overset{\infty}{\Diamond}\varphi$ then $wit^\infty(\lambda, \gamma) = \emptyset$; and if $\lambda \models \overset{\infty}{\Diamond}\varphi$ then $wit^\infty(\lambda, \gamma) = \{i \mid \lambda[i] \models \varphi\}$. The set is either infinite or empty.

Moreover, an $\mathcal{L}_{EATL^+}$ path formula $\gamma$ is called $\overset{\infty}{\Diamond}$-*atomic* if it has the form $\overset{\infty}{\Diamond}\varphi_1$. For $\varphi \in \mathcal{L}_{EATL^+}$ we denote the set of all $\overset{\infty}{\Diamond}$-atomic flat path subformulae of $\varphi$ by $\mathcal{APF}^\infty(\varphi)$.

In the following we generalise the definition of a segment.

**Definition 9.12 ($\gamma$-segment, strict).** *A* $\gamma$-segment *on a path $\lambda$ is a tuple $(i, j) \in \mathbb{N}_0^2$ with $i < j$ such that $\lambda[i] = \lambda[j]$ and for each $\gamma' \in \mathcal{APF}^\infty(\gamma)$ with $wit^\infty(\lambda, \gamma') \neq \emptyset$ there is a witness $w \in wit^\infty(\lambda, \gamma')$ such that $i \leq w \leq j$.*

*We call a $\gamma$-segment $(i, j)$* strict *if there is no other $\gamma$-segment $(k, l)$ in it.*

The next proposition shows that such $\gamma$-segments always exist on paths on which some $\overset{\infty}{\Diamond}$-atomic flat formula is true. The following proofs are done similarly to the ones given in Section 9.2.2.

**Proposition 9.13.** *Let $s_A$ be a strategy for $(\mathfrak{M}, q, \gamma)$. Then, for all paths $\lambda \in out(q, s_A)$ and $t \in \mathbb{N}$ there is a strict $\gamma$-segment $(i, j)$ on $\lambda$ with $i \geq t$.*

*Proof.* Suppose there is a path in the outcome that does not contain such a $\gamma$-segment. Then, as the set of states is finite there must be some position $l \geq t$ on $\lambda$ such that $\lambda[l, \infty]$ does not contain a witness for some $\gamma' \in \mathcal{APF}^{\infty}(\gamma)$ with $wit(\lambda, \gamma') \neq \emptyset$. But this contradicts $wit(\lambda, \gamma') \neq \emptyset$. If there is no $\overset{\infty}{\Diamond}$-formula true on a path the condition is trivially true. ■

**Lemma 9.14.** *Let $\mathfrak{M}, q \models \langle\langle A \rangle\rangle \gamma$. Then, there is a strategy $s_A$ for $(\mathfrak{M}, q, \gamma)$ such that any strict $\gamma$-segment $(i, j)$ that contains no more witnesses for any formula from $\mathcal{APF}(\gamma)$ contains at most $|Q_{\mathfrak{M}}| \cdot |\mathcal{APF}^{\infty}(\gamma)|$ states.*

*Proof.* We proceed similar to Lemma 9.5 to make all eventualities from $\mathcal{APF}(\gamma)$ true. Then, we modify the strategy to a strategy $s'_A$ such that any segment $(i_l, j_l)$ contained in any strict $\gamma$-segment $(i, j)$ contains some witness of $wit^{\infty}(\lambda, \gamma')$ for each $\gamma' \in \mathcal{APF}^{\infty}(\gamma)$ for that the witness set is non-empty on $\lambda$ (and which does not contain any more witnesses from formulae from $\mathcal{APF}(\gamma)$). Now, we consider the last segment, say $(i_l, j_l)$, contained in $(i, j)$ (i.e. $j_l = j$). If all formulae $\gamma' \in \mathcal{APF}^{\infty}(\gamma)$ with $wit^{\infty}(\lambda, \gamma') \neq \emptyset$ that have a witness in $(i_l, j_l)$ do also have a witness inside $(i, j)$ but outside $(i_l, j_l)$ then we modify $s'_A$ such that $(i_l, j_l)$ is "removed" from the $\gamma$-segment $(i, j)$ by applying the reduction of Lemma 9.5. If not, we chose the segment next to the last one and so on. The resulting $\gamma$-segment $(i, j')$ is $j_l - i_l + 1$ states shorter than $(i, j)$. Applied recursively, this procedure results in a $\gamma$-segment that contains at most $|\mathcal{APF}^{\infty}(\gamma)|$ necessary segments which are interconnected by a minimal number of states that do not contain unnecessary segments. The number of states of each segment plus the number of intermediate states between two segments is at most $|Q_{\mathfrak{M}}|$. Hence, the $\gamma$-segment contains at most $|Q_{\mathfrak{M}}|(|\mathcal{APF}^{\infty}(\gamma)|)$ states. ■

In the following we extend the finite path semantics such that it can deal with $\overset{\infty}{\Diamond}$-atomic flat formulae.

**Definition 9.15 ($\models^k$ for $\mathcal{L}_{\boldsymbol{EATL}^+}$).** *The semantics from Definition 9.6 is extended to $\mathcal{L}_{EATL^+}$-formulae by adding the following clause: $\mathfrak{M}, \lambda \models^k \overset{\infty}{\Diamond}\gamma$ iff there is some $i < k$ such that $\mathfrak{M}, \lambda[i, \infty] \models^k \gamma$.*

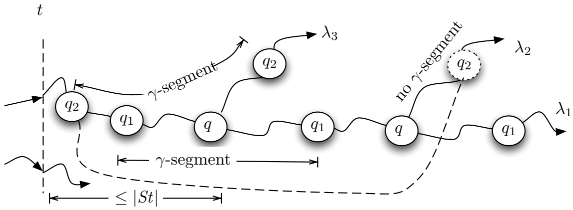



**Fig. 9.6.** Proof idea of Theorem 9.16.

The notion of a $k$-witness strategy is given analogously to Definition 9.7: $s_A$ is a *$k$-witness* strategy for $(\mathfrak{M}, q, \gamma)$ if for all $\lambda \in out(q, s_A)$ we have that $\mathfrak{M}, \lambda \models^k \gamma$.

The analog of Theorem 9.8 for $\mathbf{EATL}^+_{\mathbf{IR}}$ is given next.

**Theorem 9.16.** *We have that* $\mathfrak{M}, q \models \langle\langle A\rangle\rangle\gamma$ *iff there is a* $|Q_{\mathfrak{M}}| \cdot (1 + |\mathcal{APF}(\gamma)| + |\mathcal{APF}^\infty(\gamma)|)$*-witness strategy for* $(\mathfrak{M}, q, \gamma)$.

*Proof.* "$\Rightarrow$": Let $s_A$ be a strategy for $(\mathfrak{M}, q, \gamma)$. Then, we modify $s_A$ according to Lemma 9.5 and obtain a strategy $s'_A$ such that on all paths $\lambda$ of the outcome of $s'_A$ and for all formulae $\gamma' \in \mathcal{APF}(\gamma)$ with a witness on $\lambda$ we have that $wit(\lambda, \gamma') \leq |Q| \cdot |\mathcal{APF}(\gamma)| =: t$. We modify $s'_A$ to a strategy $s''_A$ according to Proposition 9.13 and Lemma 9.14. Finally, the states between $t$ and the start of the strict $\gamma$-segment can be shrunk up to at most $|Q|$ many, again according to Lemma 9.5 (cf. Figure 9.6) resulting in a $|Q_{\mathfrak{M}}| \cdot (1+|\mathcal{APF}(\gamma)|+|\mathcal{APF}^\infty(\gamma)|)$-witness strategy for $(\mathfrak{M}, q, \gamma)$.

"$\Leftarrow$": Now assume there is a $k := |Q_{\mathfrak{M}}| \cdot (1 + |\mathcal{APF}(\gamma)| + |\mathcal{APF}^\infty(\gamma)|)$-witness strategy for $(\mathfrak{M}, q, \gamma)$ and no strategy for $(\mathfrak{M}, q, \gamma)$. If this is caused by a formula from $\mathcal{APF}(\gamma)$, or $\gamma'$ from $\mathcal{APF}^\infty(\gamma)$ with a minimal witness position $\geq k$ the reasoning is as in the proof of Theorem 9.8. We now consider the case if it is caused by a formula from $\mathcal{APF}^\infty(\gamma)$ with a minimal witness position $< k$. Then, for any $k$-strategy $s_A$ there must be a $\gamma' \in \mathcal{APF}^\infty(\gamma)$ such that for some path $\lambda_1 \in out(q, s_A)$ it holds that $\mathfrak{M}, \lambda_1 \models^k \gamma'$ but $\mathfrak{M}, \lambda_2 \not\models \gamma'$ where $\lambda_2$ equals $\lambda_1$ up to position $k$. We show that this cannot be the case. $\mathfrak{M}, \lambda_1 \models^k \gamma'$ implies that $\gamma'$ has a witness in the initial $\gamma$-segment on $\lambda_1$ (cf. the initial $\gamma$-segment on $\lambda_1$ with start and end state $q_1$ in Figure 9.6). So, there must be a state $q$ and an outgoing path $\lambda_2$ containing no more $\gamma$-segments. However, this state and outgoing path must also be present in the initial $\gamma$-segment on the path $\lambda_1$ and on $\lambda_3$ (see Fig. 9.6) there must also be a $\gamma$-segment. If it starts within $q_1$ and $q$ on $\lambda_3$ it must also be present on $\lambda_2$. So, suppose the initial $\gamma$-segment on $\lambda_3$ with start and end state $q_2$ begins before $q_1$. But this gives us a (non-strict) $\gamma$-segment on $\lambda_2$ (shown by the dotted line) and of course, this segment can also be reached on the outgoing path $\lambda_2$ going through state $q$ on $\lambda_1$. Applying this reasoning recursively proves that each of these paths contains infinitely many $\gamma$-segments. This contradicts the assumption that $\mathfrak{M}, \lambda_2 \not\models \gamma'$. ■

The previous result allows to construct an alternating Turing machine with a fixed number of alternations to solve the model checking problem (cf. the proof of Theorem 9.9).

**Theorem 9.17.** *Let $\varphi$ be a flat $\mathcal{L}_{EATL^+}$-formula, $\mathfrak{M}$ be a* CGS, *and $q$ a state in $\mathfrak{M}$. There is a polynomial time alternating Turing machine that returns "yes" if $\mathfrak{M}, q \models_{IR} \varphi$ and "no" otherwise.*

*Proof.* The proof is done analogously to the one of Theorem 9.9. Now, the verifiers strategy and the first outcome of the opponents is constructed for the first $k := |Q_{\mathfrak{M}}| \cdot (1 + |\mathcal{APF}(\gamma)| + |\mathcal{APF}^{\infty}(\gamma)|)$ steps. Then, the game between the verifier and refuter to determine a flat atomic subformula is implemented. Finally, this subformula is tested against the guessed path regarding the semantics $\models^k$. Note, that also also the clause for $\overset{\infty}{\Diamond}$-atomic formulae has to be considered. The correctness follows from Theorem 9.16. ■

Finally, we get the following result as a combination of Theorem 9.17 and Theorem 9.2. The reasoning is exactly the same as for Theorem 9.10.

**Theorem 9.18.** *Model checking* $\mathbf{EATL}^+_{\mathbf{IR}}$ *over* CGSs *is* **PSPACE**-*complete wrt the size of the model and the length of the formula (even for turn-based models with two agents and flat* $\mathbf{ATL}^+_{\mathbf{IR}}$ *formulae).*

## 9.5 Significance of the Corrected Results

Why are the results presented here significant? First of all, we have corrected a widely believed "result" about model checking $\mathbf{ATL}^+_{\mathbf{IR}}$, and that is important on its own. Several other existing claims concerning variants of the model checking problem has been based on the "$\mathbf{\Delta}^{\mathbf{P}}_{\mathbf{3}}$-completeness" for $\mathbf{ATL}^+_{\mathbf{IR}}$, and thus needed to be rectified as well. Moreover, the $\mathbf{ATL}^+_{\mathbf{IR}}$ verification complexity is important because $\mathbf{ATL}^+_{\mathbf{IR}}$ can be seen as the minimal language discerning strategic abilities *with* and *without* memory of past actions. Our results show that the more compact models of agents (which we usually get when perfect memory is assumed) come with a computational price already in the case of $\mathbf{ATL}^+_{\mathbf{IR}}$, and not only for $\mathbf{ATL}^*_{\mathbf{IR}}$ as it was believed before.

$\mathbf{ATL}^+_{\mathbf{IR}}$ deserves attention from the conceptual point of view, too. We have argued in Section 2.2.5 that it enables neat and succinct specifications of sophisticated properties regarding e.g. the outcome of agents' play under behavioural constraints. This is especially clear for $\mathbf{EATL}^+_{\mathbf{IR}}$ where the constraints can take the form of fairness conditions. Constraints of this kind are extremely important when specifying and/or verifying agents in an asynchronous environment, cf. [Dastani and Jamroga, 2010]. Since $\mathbf{ATL}^+_{\mathbf{IR}}$ was believed to have the same model checking complexity as $\mathbf{ATL}^+_{\mathbf{Ir}}$, the former seemed a sensible tradeoff between expressivity and complexity. In this context, our new complexity results are rather pessimistic and shift the balance markedly in favour of verification of *memoryless agents*. As a consequence, for agents *with* memory one has to fall back to the less expressive logic $\mathbf{ATL}_{\mathbf{IR}}$, or accept the less desirable computational properties of $\mathbf{ATL}^+_{\mathbf{IR}}$. On the positive side, we have also shown that fairness properties incur no extra costs in either case and that model checking $\mathbf{ATL}^+_{\mathbf{IR}}/\mathbf{EATL}^+_{\mathbf{IR}}$ is still much cheaper than for $\mathbf{ATL}^*_{\mathbf{IR}}$.

# 10

# Verification of Rational Play

## Contents



In Chapters 6 and 7 we have introduced several logics to reason about rational behaviour of agents under perfect and imperfect information. In this section we analyse the complexity needed to "solve" the model checking problems for these logics. Parts of these results are based on the standard results presented in Chapter 5. We recall that the *model checking problem* refers to the question whether a given formula holds in a given model and state. We will measure the size of the input in the number of transitions in the model ($m$) and the length of the formula ($l$).

## 10.1 Rational Play under Perfect Information: ATLP

We begin with analysing the the model checking problem of **ATLP**. We show that, for different subclasses of the new logic, the complexity of model checking ranges from $\mathbf{\Delta_3^P}$-completeness to **PSPACE**-completeness. We also argue that, when the number of plausible strategy profiles is reasonably small, the model checking can be done in polynomial time. Note that the problem of checking **ATLP** with respect to the size of the *whole* CGSP (including the plausibility set $\Upsilon$), is trivially linear in the size of the model: The model size is *exponential with respect to the number of states and transitions*. Hence, model checking CGSPs does not make sense if the set of plausible strategies is stored explicitly. The set should be stored implicitly; for instance, by means of some decision procedure. We will assume throughout this section that the plausibility set $\Upsilon$ does not discriminate any strategy profiles (i.e., all strategy profiles are initially plausible), and actual plausibility assumptions must be specified in the object language through (simple or complex) plausibility terms.

The same remark applies to the denotations of primitive ("hard-wired") plausibility terms. In this respect, we will consider two subclasses of CGSPs in which the representation of plausibility assumptions of plausibility assumptions does not overwhelm the complexity of the rest of the input – namely, pure concurrent game structures and "well-behaved" CGSPs. In pure CGSs, plausibility terms and their denotations are simply absent. In well-behaved CGSs, we put a limit on the complexity of the *plausibility check*, i.e., the computational resources needed to determine whether a given strategy is plausible according to a given plausibility term and plausibility mapping.

**Definition 10.1 (CGS as CGSP, CGS-based).** *We call an* CGSP, CGS-based *if it is implicitly represented by an* CGS *in which all strategy profiles are initially plausible (i.e. $\Upsilon = \Sigma$) and there are no "hardwired" plausibility terms (i.e. $\Omega = \emptyset$).*

**Definition 10.2 (Well-behaved CGSP).** *A* CGSP $\mathfrak{M}$ *is called* well-behaved *if, and only if,*

1. *$\Upsilon_{\mathfrak{M}} = \Sigma$: all strategy profiles are plausible in $\mathfrak{M}$;*
2. *There is a non-deterministic Turing machine which decides whether $s \in [\![\omega]\!]_{\mathfrak{M}}^q$ for each state $q \in Q_{\mathfrak{M}}$, strategy profile $s \in \Sigma$, and plausibility term $\omega \in \Omega$ in polynomial time with respect to $l$ and $m$.*

*Remark 10.3.* We note that, if a list (or several alternative lists) of plausible strategy profiles is given *explicitly in the model* (via the plausibility set $\Upsilon$ and/or the denotations of abstract plausibility terms $\omega$ from Section 6.1), then the problem of guessing an appropriate strategy from such a list is in **NP** (memoryless strategies have polynomial size with respect to $m$). As a consequence, we assume that, if such a list is given explicitly, that it is stored *outside* the model.

We begin our study with the complexity of model checking the basic language $\mathcal{L}^{base}_{ATLP}$ in Section 10.1.1. Then, we consider the complexity for the intermediate language $\mathcal{L}_{ATLP^{ATLI}}$ (Section 10.1.3). It turns out that the problem is in both cases $\mathbf{\Delta_3^P}$-complete in general, which seems in line with existing results on the complexity of solving games. In particular, it is known that if both players in a 2-player imperfect information game have imperfect recall, and chance moves are allowed, then the problem of finding a max-min pure strategy is $\mathbf{\Sigma_2^P}$-complete [Koller and Megiddo, 1992].[1] That is, there are established results within game theory which show that reasoning about the outcome of a game where the strategies of both parties are restricted cannot be easier than $\mathbf{\Sigma_2^P}$ (resp. $\mathbf{\Delta_3^P}$ when nesting of game specifications is allowed). In the light of this, our complexity results are not as pessimistic as they seem, especially as **ATLP** allows specification of much more diverse restrictions than those imposed by imperfect information in 2-player turn-based games.[2]

Moreover, we show in Sections 10.1.1 and 10.1.3 that model checking $\mathcal{L}^{base}_{ATLP}$ and $\mathcal{L}^{ATLI}_{ATLP}$ is $\mathbf{\Delta_2^P}$-complete if only the proponents' strategies are restricted. This, again, corresponds to some well-known **NP**-hardness results for solving extensive games with imperfect information and recall [Chu and Halpern, 2001; Garey and Johnson, 1979; Koller and Megiddo, 1992].

Finally, in Section 10.1.4 we study the model checking complexity of $\mathcal{L}^{k}_{ATLP}$ and $\mathcal{L}_{ATLP}$. We summarise the results in Section 10.1.6.

### 10.1.1 ATLP$^{\textbf{base}}$: Upper Bounds

In this section we show that model checking $\mathcal{L}^{base}_{ATLP}$ is $\mathbf{\Delta_3^P}$-complete in general, and $\mathbf{\Delta_2^P}$-complete when only the proponents' strategies are restricted. Moreover, model checking $\mathcal{L}^{base}_{ATLP}$ over *rectangular models* and models with *bounded plausibility sets* can be done in polynomial time.

#### Well-behaved CGSPs.

A detailed algorithm for model checking $\mathcal{L}^{base}_{ATLP}$ formulae in well-behaved concurrent game structures with plausibility is presented in Figure 10.1. Apart

---

1 Note that strategic operators can be nested in an **ATLP** formula, thus specifying a sequence of games, with the outcome of each game depending on the previous ones—and solving such games requires adaptive calls to a $\mathbf{\Sigma_2^P}$ oracle.

2 In particular, imperfect information strategies (sometimes called *uniform* strategies) can be characterised in **ATLP** for a relevant subclass of models, cf. Section 10.1.2.

**function** *mcheckATLP*$(\mathfrak{M}, q, \varphi)$**;**
Model checking **ATLP**: the main function.

- Return $mcheck(\mathfrak{M}, q, \varphi, \emptyset, \emptyset)$;

**function** *mcheck*$(\mathfrak{M}, q, \varphi, \overrightarrow{\omega}, B)$**;**
Returns "true" iff $\varphi$ plausibly holds in $\mathfrak{M}, q$. The current plausibility assumptions are specified by a sequence $\overrightarrow{\omega} = [\langle\omega_1, q_1\rangle, \ldots, \langle\omega_n, q_n\rangle]$ of plausibility terms with interpretation points. The set of agents which are assumed to play rational are denoted by $B$.

cases $\varphi \equiv p$, $\varphi \equiv \neg\psi$, $\varphi \equiv \psi_1 \wedge \psi_2$ : proceed as usual;
case $\varphi \equiv (\textbf{set-pl}\ \omega')\psi$ : return( $mcheck(\mathfrak{M}, q, \psi, [\langle\omega', q\rangle], B)$);
case $\varphi \equiv (\textbf{refn-pl}\ \omega')\psi$ : return( $mcheck(\mathfrak{M}, q, \psi, \overrightarrow{\omega} \oplus \langle\omega', q\rangle, B)$);
case $\varphi \equiv \mathbf{Pl}_A\psi$ : return( $mcheck(\mathfrak{M}, q, \psi, \overrightarrow{\omega}, A)$);
case $\varphi \equiv \langle\langle A\rangle\rangle \bigcirc \psi$, where $\psi$ includes some $\langle\langle B\rangle\rangle$ : Label all $q' \in Q$, in which $mcheck(\mathfrak{M}, q, \psi, \overrightarrow{\omega}, B)$ returns "true", with a new proposition yes. Return $mcheck(\mathfrak{M}, q, \langle\langle A\rangle\rangle \bigcirc \mathsf{yes}, \overrightarrow{\omega}, B)$;
case $\varphi \equiv \langle\langle A\rangle\rangle \bigcirc \psi$, where $\psi$ includes no $\langle\langle C\rangle\rangle$ : Remove all operators **Pl**, **Ph**, (**set-pl** $\cdot$) from $\psi$ (they are irrelevant, as no cooperation modality comes further), yielding $\psi'$. Return $solve(\mathfrak{M}, q, \langle\langle A\rangle\rangle \bigcirc \psi', \overrightarrow{\omega}, B)$;
cases $\langle\langle A\rangle\rangle\Box\psi$ and $\langle\langle A\rangle\rangle\psi_1\mathcal{U}\psi_2$ : analogously ;
end case

**function** *solve*$(\mathfrak{M}, q, \varphi, \overrightarrow{\omega}, B)$**;**
Returns "true" iff $\varphi$ holds in $\mathfrak{M}, q$ under plausibility assumptions specified by $\overrightarrow{\omega}$ and applied to $B$. We assume that $\varphi \equiv \langle\langle A\rangle\rangle\Box\psi$, where $\psi$ is a propositional formula, i.e., it includes no $\langle\langle B\rangle\rangle$, **Pl**, **Ph**, (**set-pl** $\cdot$).

- Label all $q' \in Q$, in which $\psi$ holds, with a new proposition yes;
- Guess a strategy profile $s$;
- **if** $plausiblestrat(s, \mathfrak{M}, \overrightarrow{\omega}, B)$ **then** return( not $beatable(s[A], \mathfrak{M}, q, \langle\langle A\rangle\rangle\Box\mathsf{yes}, \overrightarrow{\omega}, B)$);
  **else** return( false);

**Fig. 10.1.** Model checking **ATLP** (1)

**function** $beatable(s_A, \mathfrak{M}, q, \langle\langle A\rangle\rangle\gamma, \overrightarrow{\omega}, B)$;

Returns "true" iff the opponents can beat $s_A$ so that it does not enforce $\gamma$ in $\mathfrak{M}, q$ under plausibility assumptions specified by $\overrightarrow{\omega}$ and imposed on $B$. The path formula $\gamma$ is of the form $\bigcirc\psi, \Box\psi, \psi\mathcal{U}\psi'$ with propositional $\psi, \psi'$.

- Guess a strategy profile $t$;
- **if** $plausiblestrat(t, \mathfrak{M}, \overrightarrow{\omega}, B)$ and $t|_A = s_A$ **then**
  - $\mathfrak{M}'$ := "trim" $\mathfrak{M}$, removing all transitions that cannot occur when $t|_B$ is executed;
  - return( $mcheck_{CTL}(\mathfrak{M}', q, \neg\mathsf{A}\gamma)$);

  **else** return( false);

**function** $plausiblestrat(s, \mathfrak{M}, \overrightarrow{\omega}, B)$;

Checks whether $B$'s part of strategy profile $s$ is part of some profile in $\bigcap_{\langle\omega,q\rangle\in\overrightarrow{\omega}} [\![\omega]\!]^q_{\mathfrak{M}}$.

- return true if $s|_B \in \bigcap_{\langle\omega,q\rangle\in\overrightarrow{\omega}} [\![\omega]\!]^q_{\mathfrak{M}}|_B$; and false otherwise.

**Fig. 10.2.** Model checking **ATLP** (2)

from model $\mathfrak{M}$, state $q$, and formula $\varphi$ to be checked, the input includes a plausibility specification vector $\overrightarrow{\omega}$ and a set $B$ of agents which are assumed to play rationally. The plausibility vector $\overrightarrow{\omega} = [\langle\omega_1, q_1\rangle, \dots, \langle\omega_n, q_n\rangle]$ is a sequence of plausibility terms together with states at which the terms are evaluated; this is, because we need to keep track of applications of the refinement operators (**refn-pl** $\cdot$). The intuition is that the vector represents the incremental plausibility updates. Moreover, by $[\langle\omega_1, q_1\rangle, \dots, \langle\omega_n, q_n\rangle] \oplus \langle\omega, q\rangle$ we denote the vector $[\langle\omega_1, q_1\rangle, \dots, \langle\omega_n, q_n\rangle, \langle\omega, q\rangle]$.

Since **CTL** model checking is linear in the number of transitions in the model and the length of the formula [Clarke et al., 1986] and as long as $plausiblestrat(s, \mathfrak{M}, q, \omega, B)$ can be computed in polynomial time, we get that $mcheckATLP$ runs in time $\mathbf{\Delta_3^P}$, i.e., the algorithm can be implemented as a deterministic Turing machine making adaptive calls to an oracle of range $\mathbf{\Sigma_2^P} = \mathbf{NP^{NP}}$. In fact, it suffices to require that $plausiblestrat(s, \mathfrak{M}, q, \omega, B)$ can be computed in *nondeterministic* polynomial time, as the witness for *plausiblestrat* can be guessed together with the strategy profile $s$ in function *solve*, and with the strategy profile $t$ in function *beatable*, respectively. The intersection of plausibility terms can also be neglected as the vector of plausibility terms can contain at most $l$ terms (length of the formula). Schematically, we can describe the main part of the algorithm by $\exists s \neg(\exists t)$: $s$ is guessed first, then $t$ is guessed (and its answer is negated, so we have $\exists s \forall t$). This schematic view will be useful in Section 10.1.4 to provide an intuition about the complexity of nested formulae together with quantification over strategic terms.

**Proposition 10.4.** *Let $\mathfrak{M}$ be a well-behaved* CGSP*, $q$ a state in $\mathfrak{M}$, and $\varphi$ a formula of $\mathcal{L}^{base}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \Omega)$. Then, $\mathfrak{M}, q \models \varphi$ iff mcheckATLP$(\mathfrak{M}, q, \varphi)$. The algorithm runs in time $\mathbf{\Delta^P_3}$ with respect to the number of transitions in the model and the length of the formula.*

The proof is given on page 308.

We note that the requirement that the set of plausible strategies is given by $\Sigma$ is not a real restriction. Specific plausibility specification can always be set using operator (**set-pl** $\cdot$), by adding a new plausibility term that denotes the desired set of strategy profiles. The only restriction is that inclusion in the set must be verifiable in nondeterministic polynomial time.

Finally, we observe that the complexity can be improved if only the strategies of the proponents are restricted.

**Proposition 10.5.** *Let $\gamma$ be an $\mathcal{L}^{base}_{ATLP}$-path formula without cooperation modalities. Then the model checking problem for formulae of the form $\mathbf{Pl}_A\langle\langle A\rangle\rangle\gamma$ is in $\mathbf{\Delta^P_2}$ (instead of $\mathbf{\Delta^P_3}$).*

*Proof.* We consider the case $\varphi \equiv \langle\langle A\rangle\rangle \bigcirc \psi$, where $\psi$ includes no $\langle\langle C\rangle\rangle$. In *solve* a plausible strategy $s_A$ for $A$ is guessed (**NP**-call). Then, in function *beatable* the model is directly trimmed according to $s_A$ (without guessing another profile $t$) and the **CTL** model checking algorithm is executed. In this case, function *beatable* can be executed in polynomial time. ■

**Corollary 10.6.** *Let $\varphi \in \mathcal{L}^{base}_{ATLP}$. If for each cooperation modality $\langle\langle A\rangle\rangle$ occurring in $\varphi$ it is specified that only agents $A'$ where $A' \subseteq A$ play plausibly then model checking is in $\mathbf{\Delta^P_2}$.*

*Proof.* For each cooperation modality one applies the procedure described in the proof of Proposition 10.5. ■

#### Pure CGSs and ATLP$^{\text{base}}$ without plausibility terms.

This is a somewhat degenerate case because in $\mathcal{L}^{base}_{ATLP}$ only primitive plausibility terms can be used. With no such terms, (**set-pl** $\cdot$) and (**refn-pl** $\cdot$) operators cannot be used, so all strategy profiles will be considered plausible in the evaluation of every subformula. In consequence, model checking $\mathcal{L}^{base}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \emptyset)$ can be done in the same way as for **ATL**. Since model checking **ATL** lies in **P** [Alur et al., 2002] we get the following result.

**Proposition 10.7.** *Let* $\mathfrak{M}$ *be a* CGSP, $q$ *a state in* $\mathfrak{M}$, *and* $\varphi \in \mathcal{L}_{ATLP}^{base}(\mathbb{A}\mathrm{gt}, \Pi, \emptyset)$. *Model checking* $\varphi$ *in* $\mathfrak{M}, q$ *is in* **P** *with respect to the number of transitions in the model and the length of the formula.*

*Proof.* Remove all $\mathbf{Pl}_A$ operators from $\varphi$ and check whether $\mathfrak{M}', q \models_{\mathsf{ATL}} \varphi$ where $\mathfrak{M}'$ is the CGS obtained from $\mathfrak{M}$ by leaving out $\Upsilon, \Omega$, and $[\![\cdot]\!]$. ■

**Special Classes of Models.**

We will now consider the special case in which each plausibility term refers to at most polynomially many strategies.

**Definition 10.8 (Bounded models $\mathfrak{M}^c$).** *Given a fixed constant* $c \in \mathbb{N}$ *we consider the class* $\mathfrak{M}^c \subseteq CGSP(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ *of models such that for all* $\mathfrak{M} \in \mathfrak{M}^c$, $\omega \in \Omega_{\mathfrak{M}}$, *and* $q \in Q_{\mathfrak{M}}$ *it holds that* $|[\![\omega]\!]_{\mathfrak{M}}^q| \leq l^c \cdot m^c$ *where* $l$ *(resp.* $m$*) denotes the length of the input formula (resp. number of transitions of* $\mathfrak{M}$*).*

**Proposition 10.9.** *Let* $c \in \mathbb{N}$ *be a constant. Model checking* $\mathcal{L}_{ATLP}^{base}$ *formulae with respect to the class of well-behaved bounded models* $\mathfrak{M}^c$ *can be done in polynomial time with respect to the number of transitions in the model and the length of the formula.*

*Proof.* [Idea] The idea is to apply **ATL** model checking for any of the polynomially many plausible strategies. The complete proof is given on page 309. ■

Even with arbitrarily many strategies the complexity can be improved if the set of plausible profiles has a specific structure, namely if the set can be (and is) represented in a *rectangular* way. Intuitively, such a set of profiles can be represented by behavioural constraints [van der Hoek et al., 2005b]. That is, we restrict the actions that can be performed independently for each state and agent, and then consider all strategy profiles generated from the constrained repertoire of actions.

**Definition 10.10 (Rectangularity, $\mathfrak{M}^{\mathbf{rect}}$).** *Let* $S_a \subseteq \Sigma_a$ *be a set of strategies of agent* $a$. *We say that* $S_a$ *is* rectangular *if it is represented by a function* $d'_a : Q_{\mathfrak{M}} \to \mathcal{P}(Act) \backslash \{\emptyset\}$ *such that for all states* $q \in Q_{\mathfrak{M}}$ *it holds that* $d'_a(q) \subseteq d_a(q)$; *then,* $S_a$ *is taken to be the set* $\{s_a \in \Sigma_a \mid \forall q \in Q_{\mathfrak{M}}\ (s_a(q) \in d'_a(q))\}$.

*A set of collective strategies (resp. strategy profiles)* $S_A \subseteq \Sigma_A$ *is rectangular if it represented as a collection of rectangular sets of individual strategies. Then,* $S_A$ *is to the Cartesian product of the individual sets, i.e.,* $S_A = \times_{a \in A} S_a$.

*A set of plausibility terms* $\Omega$ *is rectangular in a model* $\mathfrak{M}$ *if all terms in* $\omega \in \Omega$ *have rectangular denotations* $[\![\omega]\!]_{\mathfrak{M}}^q$. *Finally, we say that a* CGSP $\mathfrak{M}$ *is rectangular if the set* $\Upsilon_{\mathfrak{M}}$ *is rectangular and terms* $\Omega$ *are rectangular in* $\mathfrak{M}$. *We denote the class of such models by* $\mathfrak{M}^{rect}$.

Note, for example, that each $\Sigma_A$ is rectangular.

**Proposition 10.11.** *Model checking $\mathcal{L}_{ATLP}^{base}$ formulae in the class $\mathfrak{M}^{rect}$ can be done in* **P** *with respect to the number of transitions in the model and the length of the formula.*

*Proof.* The algorithm is very simple; we present the procedure for $\varphi \equiv \langle\langle A\rangle\rangle\Box\psi$ being in the scope of (**set-pl** $\omega$) and **Pl** $_B$. Other cases are analogous.

First, we model-check (**set-pl** $\omega$)**Pl** $_B\psi$ recursively and label the states where the answer was "true" with a new proposition $\mathsf{yes}$. Then, we take $[\![\omega]\!]_{\mathfrak{M}}^q$ (recall that it is represented in a rectangular way, i.e., by function $d' : \mathbb{A}\mathrm{gt} \times Q \to \mathcal{P}(Act)$), and replace function $d$ in $\mathfrak{M}$ by $d''$ such that $d''(a,q) = d'(a,q)$ for $a \in B$ and $d''(a,q) = d(a,q)$ for $a \notin B$. Finally, we use any **ATL** model checker to model-check $\langle\langle A\rangle\rangle\Box\mathsf{yes}$ in the resulting model, and return the answer. ■

We observe that strategic combinations of rectangular plausibility terms are also rectangular. AS a consequence, the results extends to $\mathcal{L}_{ATLP}^{0}$ in a straightforward way, which proves to be useful in Section 10.1.4.[3]

**Lemma 10.12.** *If $S \subseteq \Sigma_a$ (resp. $S \subseteq \Sigma_A$) contains only a single strategy (resp. strategy profile) then it is rectangular.*

*Proof.* We just take $d'_a(q)$ as $\{s_a(q) \mid s_a \in S\}$ for all $q \in Q$. ■

**Lemma 10.13.** *Let $\Omega$ be a rectangular set of plausibility terms, then $\mathcal{T}(\Omega)$ is rectangular as well.*

*Proof.* Let $\Omega = \{\omega_1, \dots, \omega_w\}$. Each plausibility term is rectangular; let it be represented by $\omega_i = S_1^i \times \dots S_k^i$. Then, we have that

$$[\![(\omega_{i_1}, \dots, \omega_{i_k})]\!] = \{s \mid \exists t^{i_j} \in [\![\omega_{i_j}]\!] : (t^{i_j}|_{i_j} = s|_{i_j}\} = \\ = S_{i_1}^1 \times S_{i_2}^2 \times \dots \times S_{i_k}^k$$

and

$$[\![\omega_i[A]]\!] = \{s \mid \exists t \in [\![\omega_i]\!] : (t|_A = s|A)\} = \\ = \{s \mid \forall a \in A \exists t^a \in [\![\omega_i]\!] : (t^a|_a = s|_a)\} = \\ = \{s \mid \forall a \in A : (s|_a \in S_a^i) \wedge \forall a \in \mathbb{A}\mathrm{gt}\backslash A : (s_a \in \Sigma_a)\}.$$

Both sets are rectangular. ■

The following corollary is immediate from Proposition 10.11 and the previous lemmata since the set of plausibility terms in $\mathcal{L}_{ATLP}^{0}$ are described by the $\mathcal{T}$-operator.

---

[3] Recall, that $\mathcal{L}_{ATLP}^{0}$ consists of all base formulae in which plausibility terms form $\mathcal{T}(\Omega)$ can be used (instead of plain terms from $\Omega$ only).

**Corollary 10.14.** *Model checking $\mathcal{L}^0_{ATLP}$ formulae in the class $\mathfrak{M}^{rect}$ can be done in* **P** *with respect to the number of transitions in the model and the length of the formula.*

### 10.1.2 $\mathsf{ATLP}^{\mathsf{base}}$: Hardness and Completeness

**Well-behaved** CGSPs**.**

We prove $\mathbf{\Delta_3^P}$-hardness through a reduction of $\mathrm{SNSAT}_2$ from Definition 4.15, a typical $\mathbf{\Delta_3^P}$-complete variant of the Boolean satisfiability problem. The reduction is done in two steps.

1. Firstly, we define a modification of $\mathsf{ATL_{ir}}$ [Schobbens, 2004] which we recall in Section A.3, in which *all* agents are required to play only uniform strategies. We call it "uniform $\mathsf{ATL_{ir}}$" ($\mathsf{ATL}^u_{\mathsf{ir}}$ in short), and show that model checking $\mathsf{ATL}^u_{\mathsf{ir}}$ is $\mathbf{\Delta_3^P}$-complete by means of a polynomial reduction of $\mathrm{SNSAT}_2$ to $\mathsf{ATL}^u_{\mathsf{ir}}$ model checking.
2. Then, we point out that each *relevant* formula and model of $\mathsf{ATL}^u_{\mathsf{ir}}$ can be equivalently translated (in polynomial time) to a CGSP and an $\mathcal{L}^{base}_{ATLP}$-formula, thus yielding a polynomial reduction of $\mathrm{SNSAT}_2$ to model checking $\mathsf{ATLP}^{\mathsf{base}}$.

Parts of our construction reuse techniques presented in [Goranko and Jamroga, 2004; Jamroga and Dix, 2006; Jamroga, 2007; Jamroga and Dix, 2008].

In "uniform $\mathsf{ATL_{ir}}$" ($\mathsf{ATL}^u_{\mathsf{ir}}$) (cf. Section A.3) we assume that all the players have limited information about the current state, and each agent can only use *uniform* strategies (i.e., ones that assign same choices in indistinguishable states). The syntax of $\mathsf{ATL}^u_{\mathsf{ir}}$ is the same as that of $\mathsf{ATL}$. The semantics of $\mathsf{ATL}^u_{\mathsf{ir}}$ is defined over ICGS (cf. Definition 2.27), i.e. CGS extended with epistemic relations that represent indistinguishability of states for agents. Again, details of the semantics and more thorough presentation can be found in Appendix A.3. The following proposition summarises the complexity results from Appendix B.3.

**Theorem 10.15.** *Model checking* $\mathsf{ATL}^u_{\mathsf{ir}}$ *is* $\mathbf{\Delta_3^P}$*-complete with respect to the number of transitions in the model and the length of the formula.*

The complete proof is given on page 307 and Section B.3.

*Remark 10.16.* We have thus proven that checking strategic abilities when *all* players are required to play uniformly is $\mathbf{\Delta_3^P}$-complete. We believe it is an interesting result with respect to verification of various kinds of agents' abilities under incomplete information. We note that the result from [Koller and Megiddo, 1992] for extensive games with incomplete information can be seen as a specific case of our result, at least in the class of games with binary payoffs.

Now we show how $\mathsf{ATL}^u_{\mathsf{ir}}$ model checking can be reduced to model checking $\mathsf{ATLP}^{\mathsf{base}}$. We are given an ICGS $\mathfrak{M}$, a state $q$ in $\mathfrak{M}$, and an $\mathsf{ATL}^u_{\mathsf{ir}}$ formula $\varphi$. Let $\Sigma^u$ be the set of all uniform strategy profiles in $\mathfrak{M}$. We take CGSP $\mathfrak{M}'$ as $\mathfrak{M}$ (without epistemic relations) extended with plausibility mapping $[\![\cdot]\!]$ such that $[\![\omega]\!]^q = \Sigma^u$. Then, we would like to have that $\mathfrak{M}, q \models_{\mathsf{ATL}^u_{\mathsf{ir}}} \langle\!\langle A \rangle\!\rangle \varphi$ if, and only if, $\mathfrak{M}', q \models_{\mathsf{ATLP}} (\mathbf{set\text{-}pl}\ \omega)\mathbf{Pl}\,\langle\!\langle A \rangle\!\rangle \varphi$ which would complete the reduction. Unfortunately, in general this is not the case as $\mathsf{ATL}^u_{\mathsf{ir}}$ requires a winning strategy that is successful in *all* states indistinguishable from the current one (cf. Section A.3). However, we can show the following result.

**Theorem 10.17.** *Let $\mathfrak{M}$ be a* ICGS*, $q$ a state in $\mathfrak{M}$ that is only indistinguishable from itself, and $\langle\!\langle A \rangle\!\rangle \gamma$ be an $\mathsf{ATL}^u_{\mathsf{ir}}$ formula such that $\gamma$ is flat (i.e. does not contain any more cooperation modalities). Let $\Sigma^u$ be the set of all uniform strategy profiles in $\mathfrak{M}$. We take* CGSP *$\mathfrak{M}'$ as $\mathfrak{M}$ (without epistemic relations) extended with plausibility mapping $[\![\cdot]\!]$ such that $[\![\omega]\!]^q = \Sigma^u$. Then, we have that*

$$\mathfrak{M}, q \models_{\mathsf{ATL}^u_{\mathsf{ir}}} \langle\!\langle A \rangle\!\rangle \gamma \text{ if, and only if, } \mathfrak{M}', q \models_{\mathsf{ATLP}} (\textbf{\textit{set-pl}}\ \omega)\mathbf{Pl}\,\langle\!\langle A \rangle\!\rangle \gamma.$$

*Proof.*

$\mathfrak{M}, q \models_{\mathsf{ATL}^u_{\mathsf{ir}}} \langle\!\langle A \rangle\!\rangle \gamma$

iff there is a uniform strategy $s_A$ that that for each uniform counterstrategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$ and $\lambda \in out([q]_A, (s_A, t_{\mathbb{A}\mathrm{gt}\setminus A})$ we have that $\mathfrak{M}, \lambda \models_{\mathsf{ATL}^u_{\mathsf{ir}}} \gamma$

iff there is a uniform strategy $s_A$ that that for each uniform counterstrategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$ and $\lambda \in out(q, (s_A, t_{\mathbb{A}\mathrm{gt}\setminus A})$ we have that $\mathfrak{M}, \lambda \models_{\mathsf{ATL}^u_{\mathsf{ir}}} \gamma$

iff there is a plausible strategy $s_A \in [\![\omega]\!]^q$ that that for every $\lambda \in out_{\mathfrak{M}''}(q, s_A, \mathbb{A}\mathrm{gt})$ we have that $\mathfrak{M}'', \lambda \models_{\mathsf{ATLP}} \gamma$

iff $\mathfrak{M}', q \models_{\mathsf{ATLP}} (\mathbf{set\text{-}pl}\ \omega)\mathbf{Pl}\,\langle\!\langle A \rangle\!\rangle \gamma$

where $\mathfrak{M}''$ equals $\mathfrak{M}'$ but the set of plausible strategy profiles in the model is given by $\Sigma^u$. ■

*Remark 10.18.* We note in passing that, technically, the size of the resulting model $\mathfrak{M}'$ is not entirely polynomial. $\mathfrak{M}'$ includes the plausibility set $\Upsilon$, which is exponential in the number of states in $\mathfrak{M}$ (since it is equal to the the set of all uniform strategy profiles in $\mathfrak{M}$). This is of course the case when we want to store $\Upsilon$ explicitly. However, checking if a strategy profile is uniform can be done in time linear wrt the number of states in $\mathfrak{M}$, so an *implicit* representation of $\Upsilon$ (e.g., the checking procedure itself) requires only linear space.

A closer analysis of the models which are obtained from the SNSAT$_2$ reduction to $\mathbf{ATL}_{\mathbf{ir}}^{u}$ shows that the resulting ICGSs have a very specific structure.

Let $M_{\mathrm{SNSAT}_2}$ contain the models that are obtained by the construction shown in Section B.3 for any SNSAT$_2$ instance. Then, we have the following result.

**Proposition 10.19.** *For each $\mathfrak{M} \in M_{\mathrm{SNSAT}_2}$ only literal states are connected via indistinguishability relations.*

*Moreover, given an* SNSAT$_2$ *instance with $r = 1, \ldots, p$ let $\mathfrak{M}$ and $\Phi_r$, $r = 1, \ldots, p$ be the model and formulae constructed according to Section B.3 and Proposition B.2, respectively. Then, formulae $\Phi_r$ are only evaluated in states $q$ where $q$ is the only state indistinguishable from $q$.*

*Proof.* The first point is immediate from the construction. The latter result is proven by induction on $r$.

Let $r = 1$. According to Definition we have that

$$z_1 \equiv \exists X_1^1 \forall X_1^2 \exists X_1^3 \ldots Q X_1^i . \varphi_1(X_1^1, \ldots, X_1^i).$$

The quantifiers do not affect the structure of the ICGS. If $\varphi_1 = l$ for some literal $l$ the claim is trivially true. For all other cases the formula $\Phi_1$ is evaluated in the initial state which represents the outermost (wrt infix notation) Boolean connective of the formula $\varphi_1$. Such states are only indistinguishable to themselves by construction.

For $r = p$ the outermost (i.e. the one on the left evaluated next) cooperation modality of $\Phi_p$ is evaluated in the initial state; the claim follows as for $r = 1$. Nested cooperation modalities are only evaluated in states labelled $\mathsf{neg}$ and only states corresponding to $\neg z_i$ for some $i = 1, \ldots, p - 1$ have such labels. The result follows from the construction as these states are only indistinguishable to themselves. ■

Finally, we obtain the following theorem.

**Theorem 10.20.** *Model checking $\mathcal{L}_{ATLP}^{base}$ for well-behaved* CGSPs *is $\mathbf{\Delta_3^P}$-complete with respect to the number of transitions in the model and the length of the formula.*

*Proof.* Membership in $\mathbf{\Delta_3^P}$ follows from Proposition 10.4. For the hardness we reduce SNSAT$_2$ (resp. $\mathbf{ATL}_{\mathbf{ir}}^{u}$ over the restricted class of models discussed above) to **ATLP** by means of Proposition 10.19, Theorem 10.17 and Theorem 10.15. In particular, we get the following reduction by Proposition B.2:

$$z_p \text{ iff } \mathfrak{M}_p, q_0^p \models_{\mathbf{ATL}^u_{\mathbf{ir}}} \Phi_p \text{ iff } \mathfrak{M}', q_0^p \models_{\mathbf{ATLP}} (\textbf{set-pl}\ \omega)\mathbf{Pl}\,\Phi_p$$

where $\mathfrak{M}'$ is the CGSP defined as $\mathfrak{M}_p$ (without epistemic relations) in which the set of plausible strategies is given by all uniform strategies in $\mathfrak{M}_p$. ■

For the special case when only the proponents have to follow plausible strategies, a reduction from model checking $\mathbf{ATL_{ir}}$ (instead of $\mathbf{ATL^u_{ir}}$) is sufficient. Since model checking $\mathbf{ATL_{ir}}$ is $\mathbf{\Delta_2^P}$-complete [Schobbens, 2004; Jamroga and Dix, 2008], we get the following result.

**Corollary 10.21.** *Let $\mathcal{L}$ the subset of $\mathcal{L}^{base}_{ATLP}$ in which each cooperation modality $\langle\langle A\rangle\rangle$ occurs in the scope of $\mathbf{Pl}_B$ with $B \subseteq A$. Then, model checking $\mathcal{L}$ in the class of well-behaved* CGSPs *is $\mathbf{\Delta_2^P}$-complete.*

*Proof.* The inclusion in $\mathbf{\Delta_2^P}$ has been already shown in Section 10.1.1. We prove the lower bound by a reduction of model checking Schobbens' $\mathbf{ATL_{ir}}$ [Schobbens, 2004] (cf. Theorem 5.8) to model checking of our sublanguage $\mathcal{L}$. More precisely, we consider $\mathbf{ATL_{ir}}$ over a restricted class of models: The class which is obtained by encoding $\text{SNSAT}_1$ as ICGSs. Technically, the reduction is similar to the one given in Section B.3. We obtain an analogous result of Proposition 10.19. Also a similar theorem to Theorem 10.17 holds we only have to use $\mathbf{Pl}_{\mathbf{v}}$ instead of $\mathbf{Pl}$ (i.e. only the verifier is forced to use uniform strategies). This gives us a reduction of $\text{SNSAT}_1$ to model checking $\mathcal{L}$ following the same reasoning as in the proof of Theorem 10.20:

$$z_p \text{ iff } \mathfrak{M}_p, q_0^p \models_{\mathbf{ATL}^u_{\mathbf{ir}}} \Phi_p \text{ iff } \mathfrak{M}', q_0^p \models_{\mathbf{ATLP}} (\textbf{set-pl}\ \omega)\mathbf{Pl}_{\mathbf{v}}\Phi_p.$$

■

**Pure** CGS **and Special Classes of Models.**

In order to show lower bounds for model checking $\mathcal{L}^{base}_{ATLP}$ for pure concurrent game structures, well-behaved bounded models, and rectangular models, we observe that $\mathbf{ATL}$ is a subset of $\mathcal{L}^{base}_{ATLP}$ even if the latter does not use plausibility terms at all–and model checking $\mathbf{ATL}$ is $\mathbf{P}$-complete [Alur et al., 2002]. Thus, we conclude with the following thanks to Proposition 10.11.

**Theorem 10.22.** *Let $c \in \mathbb{N}$ be a constant. Model checking $\mathcal{L}^{base}_{ATLP}$ with respect to well-behaved bounded models $\mathfrak{M}^c$, rectangular models $\mathfrak{M}^{rect}$, and pure* CGSs *is $\mathbf{P}$-complete.*

### 10.1.3 $\mathbf{ATLP^{ATLI}}$

Here, we show that model checking $\mathbf{ATLP^{ATLI}}$ is also $\mathbf{\Delta_3^P}$-complete. Note that the only primitive terms occurring in $\mathcal{L}^{ATLI}_{ATLP}$-formulae are used to simulate strategic terms of $\mathbf{ATLI}$ (which denote individual strategies of particular agents).

### Upper Bound

The algorithm in Figure 10.6 uses abstract plausibility terms but it can also be used for $\mathcal{L}_{ATLI}$-based plausibility terms presented in Section 6.2.1. In [Jamroga et al., 2005] it was shown that the model checking problem for **ATLI** is polynomial with respect to the number of transitions and length of the formula. Thus, we get another immediate corollary of Proposition 10.4.

**Proposition 10.23.** *Model checking* $\mathcal{L}_{ATLP}^{ATLI}$ *in well-behaved* CGSPs *is in* $\mathbf{\Delta_3^P}$ *with respect to the number of transitions in the model and the length of the formula.*

In Section 6.2.2 we have used $\mathcal{L}_{ATLP}^{1}$ formulae to characterise game theoretic solution concepts. For this purpose it was not necessary to have hardwired plausibility terms in the language. Indeed, the absence of such terms positively influences the model checking complexity of higher levels of **ATLP**.

### Hardness and Completeness

As in Section 10.1.2, we show the lower bound by a reduction from model checking $\mathbf{ATL}_{\mathbf{ir}}^{u}$. That is, we demonstrate how uniformity of strategy profiles can be characterised by $\mathcal{L}_{ATLI}$-formulae for a relevant class of concurrent game structures. The actual reduction is quite technical and can be found in Appendix B.4.3. The following result is an immediate consequence of Proposition B.6, presented in Appendix B.4.3.

**Theorem 10.24.** *Model checking* $\mathbf{ATLP}^{\mathbf{ATLI}}$ *in well-behaved* CGSPs *is* $\mathbf{\Delta_3^P}$*-complete with respect to the number of transitions in the model and the length of the formula.*

Moreover, if plausibility restrictions apply only to proponents, then the complexity improves (the proof is done analogously to Corollary 10.21).

**Theorem 10.25.** *Let* $\mathcal{L}$ *the subset of* $\mathcal{L}_{ATLP}^{ATLI}$ *in which every cooperation modality* $\langle\langle A\rangle\rangle$ *occurs in the scope of* $\mathbf{Pl}_B$ *with* $B \subseteq A$*. Then, model checking* $\mathcal{L}$ *in the class of well-behaved rectangular* CGSPs *is* $\mathbf{\Delta_2^P}$*-complete.*

*Proof.* [sketch] We prove the lower bound (again) by a reduction of model checking $\mathbf{ATL}_{\mathbf{ir}}$ to model checking $\mathcal{L}$. The reduction is very similar to the one shown in Appendix B.4.3 except that only the "verifier" decides upon the values of the propositions (cf. [Jamroga and Dix, 2006]). ∎

### 10.1.4 **ATLP**: Upper Bounds

In this section we present our results regarding the model checking complexity of the full logic **ATLP**. The complexity depends on both the nesting level of $\mathcal{L}_{ATLP}$-formulae and on the structure and alternations of strategic quantifiers. Before we state our results we introduce some additional definitions needed to classify such complex formulae.

#### Classifying $\mathcal{L}_{ATLP}$-Formulae: Some Definitions

The complexity of model checking $\mathcal{L}_{ATLP}$-formulae does not only depend on the actual nesting depth of plausibility terms but also on the structure of strategic quantifiers used inside (**set-pl** $\cdot$) and (**refn-pl** $\cdot$) operators. The latter structure is quite complex and cannot solely be described by the number of quantifiers. Often, a specific position of quantifiers can be used to combine two "guessing" phases, improving complexity.

Firstly, not the number of quantifiers is important but rather the number of alternations. We introduce function $\textsc{Alt} : \{\exists, \forall\}^+ \to \{\exists, \forall\}^+$ which modifies a word over $\{\exists, \forall\}$ such that each quantifier following a quantifier of the same type is removed; for example, $\textsc{Alt}(\exists\forall\forall\forall\exists\forall) = \exists\forall\exists\forall$. Moreover, existential quantifiers at the beginning of a quantifier series can, under some conditions, be ignored without changing the model checking complexity. If the first quantifier is existential it follows the guess of the proponents (resp. opponents) strategy and both guesses can be combined. To take this into account, we define function $\text{R}\textsc{Alt} : \{\exists, \forall\}^+ \to \mathbb{Z}$ that counts the number of *relevant alternations of quantifiers* in a sequence:

$$\text{R}\textsc{Alt}(\overrightarrow{Q}) = \begin{cases} n & \text{if } \textsc{Alt}(\overrightarrow{Q}) = Q_1 \dots Q_n \text{ and } Q_1 \neq \exists; \\ n-1 & \text{if } \textsc{Alt}(\overrightarrow{Q}) = Q_1 \dots Q_n \text{ and } Q_1 = \exists; \end{cases}$$

Function $\text{R}\textsc{Alt}$ characterises the "hardness" of the outermost level in a given term. The next two functions take into account the recursive structure of terms, due to possibly nested (**set-pl** $\cdot$) or (**refn-pl** $\cdot$) operators. First, $\mathcal{UO}(\varphi)$ returns the set of all the *update operations* (**set-pl** $\omega$) and (**refn-pl** $\omega$) within formula $\varphi$. Second, *ql* takes a set of update operations and returns the *quantifier level* in these operations as follows:

$$ql(S) = \begin{cases} \max_{s \in S} ql(\{s\}) & \text{if } |S| > 1, \\ ql(\mathcal{UO}(\varphi')) & \text{if } S = \{(\mathbf{Op}\ \sigma.\varphi')\}, \\ \text{R}\textsc{Alt}(Q_1 \dots Q_n) + ql(\mathcal{UO}(\varphi')) & \text{if } S = \{(\mathbf{Op}\ \sigma.Q_1\sigma_1 \dots Q_n\sigma_n\varphi')\}, \\ 0 & \text{if } S = \emptyset \text{ or} \\ & (S = \{(\mathbf{Op}\ \omega)\} \text{ with } \omega \in \mathcal{T}(\Omega \cup \mathcal{V}ar)), \end{cases}$$

where (**Op** ·) is either (**set-pl** ·) or (**refn-pl** ·).

*Remark 10.26.* The *ql* and RALT operators in [Bulling et al., 2009b] were flawed and have been corrected.

The intuition behind *ql* is that it determines the maximal sum of relevant alternations in each sequence of nested update operators (**set-pl** ·), (**refn-pl** ·). Intuitively, the nested operators represent a tree. Given an $\mathcal{L}^k_{ATLP}$-formula we add arcs from the root of the tree to nodes representing update operators on the $k$th level. Then, from such a new node representing (**set-pl** $\omega$) or (**refn-pl** $\omega$), we add arcs to nodes representing update operators inside $\omega$ (i.e., on the $k-1$th level) and so on. Leaves of the tree consist of nodes representing operators whose terms contain no further update operators. Now, each node represented by e.g. (**set-pl** $\sigma.Q_1\sigma_1 \dots Q_n\sigma_n\varphi'$) is labeled by RALT$(Q_1 \dots Q_n)$. Function *ql* returns the maximal sum of such numbers along all paths from the root to some leaf.

**Definition 10.27 (Level $i$ formula).** *We say that $\varphi$ is a* level $i$ formula *iff* $ql(\mathcal{UO}(\varphi)) = i$.

*Example 10.28.* Consider the following $\mathcal{L}^2_{ATLP}$-formula:

$$\varphi \equiv (\textbf{set-pl}\ \sigma.\forall\sigma_1\exists\sigma_2\exists\sigma_3(\textbf{set-pl}\ \sigma.\forall\sigma'_1\exists\sigma'_2\exists\sigma'_3\forall\sigma'_4\varphi''))\textbf{Pl}\,\langle\!\langle A\rangle\!\rangle \bigcirc \mathsf{p}.$$

This formula is a level-5 formula. We have that

$$\mathcal{UO}(\varphi) = \{(\textbf{set-pl}\ \sigma.\forall\sigma_1\exists\sigma_2\exists\sigma_3(\textbf{set-pl}\ \sigma.\forall\sigma'_1\exists\sigma'_2\exists\sigma'_3\forall\sigma'_4)\varphi'')\}$$

and $ql(\mathcal{UO}(\varphi)) = \text{RALT}(\forall\exists\exists) + ql(\mathcal{UO}(S)) = 2 + 3 = 5$ where $S = \{(\textbf{set-pl}\ \sigma.\forall\sigma'_1\exists\sigma'_2\exists\sigma'_3\forall\sigma'_4)\varphi''\}$ and $ql(\mathcal{UO}(S)) = \text{RALT}(\forall\exists\exists\forall) + ql(\mathcal{UO}(\varphi'')) = 3 + 0$ where $ql(\mathcal{UO}(\varphi'')) = 0$ as $\varphi$ is a $\mathcal{L}^2_{ATLP}$-formula.

### Upper Bounds

Plausibility terms are quite important for the base language $\mathcal{L}^{base}_{ATLP}$; it does not make much sense to consider the logic without them. In fact, when $\mathcal{L}^{base}_{ATLP}$-formulae are considered in the context of pure CGSs, the whole logic degenerates to pure **ATL**. This observation does not apply to higher levels of **ATLP** any more. Indeed, all characterisations of game theoretic solutions concepts that we have presented are expressed as $\mathcal{L}^1_{ATLP}$-formulae *without* hard-wired terms. Moreover – as we shall see – not using hard-wired terms yields an improved model checking complexity.

Below we state the main results of this section. The intuition is the following. For each level $i$ formula we have $i$ quantifier alternations; in addition to that, in each level there can be two more implicit quantifiers due to the cooperation modalities (*there is* a plausible strategy of the proponents such that *for all* plausible strategies of the opponents ...).

**Theorem 10.29 (Model checking $\mathcal{L}^k_{ATLP}$ in pure CGSs).** *Let $\varphi$ be a level-i formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \emptyset)$, $k \geq 1$, $i \geq 0$. Moreover, let $\mathfrak{M}$ be a CGS, and $q$ a state in $\mathfrak{M}$. Then, model checking $\mathfrak{M}, q \models \varphi$ can be done in time $\mathbf{\Delta^P_{i+2k+1}}$.*

The complete proof is given on page 310.

Note, that the restriction to pure CGSs is essential because defining a given set of strategies $\Upsilon$ might require checking whether a strategy is plausible in the final nesting stage. In this case the advantage of not having hard-wired plausibility terms would vanish and the complexity would increase. So, if plausibility terms are available the last level of an $\mathcal{L}_{ATLP}$-formula cannot be verified in polynomial time anymore (according to Corollary 10.14). The complexity can increase as shown in the following result.

**Theorem 10.30 (Model checking $\mathcal{L}^k_{ATLP}$ in well-behaved CGSPs).** *Let $\varphi$ be a level-i formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$, $\mathfrak{M}$ a well-behaved CGSP, and $q$ a state in $\mathfrak{M}$. Model checking $\mathfrak{M}, q \models \varphi$ can be done in $\mathbf{\Delta^P_{i+2(k+1)+1}}$.*

The complete proof is given on page 312.

*Remark 10.31.* As a consequence of Remark 10.26 we have corrected the previous two theorem stated in [Bulling et al., 2009b].

### 10.1.5 ATLP: Hardness and Completeness

As it turns out, model checking **ATLP**, and even each **ATLP**$^k$ for $k \geq 1$ is in general **PSPACE**-complete. To show the lower bounds for **ATLP**$^k$ (with arbitrary $k \geq 1$) we show that **ATLP**$^1$ is **PSPACE**-hard, implying that all logics $\mathcal{L}^k_{ATLP}$ (for $k \geq 1$) are **PSPACE**-hard too. That the general model checking problem for **ATLP** formulae is in **PSPACE** follows directly from the algorithm shown in Figure 10.6.

The hardness proof, similar to the one for **ATLP**$^{\mathbf{ATLI}}$. We use *quantified satisfiability* (QSAT) to show **PSPACE**-completeness of model checking $\mathcal{L}^k_{ATLP}$ and $\mathcal{L}_{ATLP}$.

Given an instance $\varphi$ of QSAT we construct an $\mathcal{L}^1_{ATLP}$ formula $\theta_\varphi$ and a CGSP $\mathfrak{M}_\varphi$ (both are constructible in polynomial space regarding the length of $\varphi$) such that $\varphi$ is satisfiable if, and only if, $\mathfrak{M}_\varphi, q_0 \models \theta_\varphi$. In the following we sketch the constructions which are based on the reduction of $\mathrm{SNSAT}_2$ to model checking $\mathbf{ATL}^u_{\mathbf{ir}}$ proposed in Appendix B.3, and the translation of $\mathbf{ATL}^u_{\mathbf{ir}}$ to $\mathcal{L}_{ATLP^{ATLI}}$ proposed in Appendix B.4.3.

Let $\varphi \equiv \exists x_1 \forall x_2 \ldots Q_n x_n\, \psi$ be an instance of QSAT. Firstly, we sketch the construction of the ICGS $\mathfrak{M}'_\varphi$ which will then be transformed into a CGSP $\mathfrak{M}_\varphi$. In comparison to the construction in Appendix B.3, we consider $n$ agents one for each quantifier (in fact, we consider $\max\{2, n\}$ agents; however, for the rest of this section we assume that $n \geq 2$). The agent belonging to quantifier

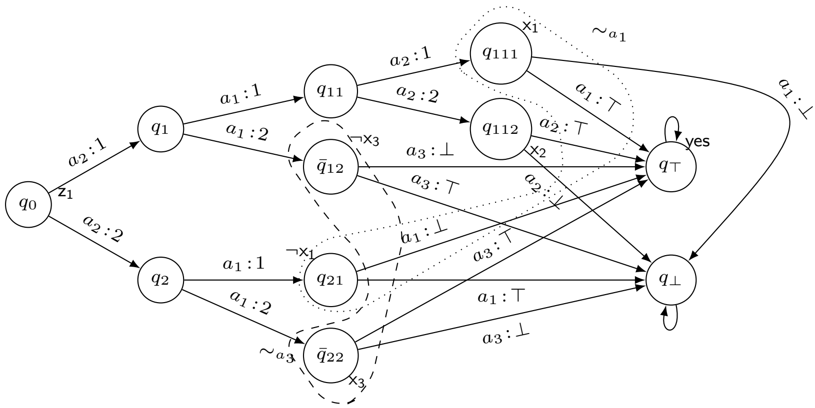



**Fig. 10.3.** Construction of the intermediate model $\mathfrak{M}'_\varphi$ for $\varphi \equiv \exists x_1 \forall x_2 \exists x_3((x_1 \wedge x_2) \vee \neg x_3) \wedge (\neg x_1 \vee x_3)$.

$i$ is named $a_i$. Except for the proposition states the procedure is completely analogous to the construction given in Appendix B.3 where agent $a_2$ acts as *refuter* and $a_1$ as *verifier*. (Alternatively, two additional agents could be added.) The procedure at the proposition states changes as follows: In such a state, say $q$, referring to a literal $l$, say $l = x_i$, agent $a_i$ can decide on the value of $x_i$. Note again that the agent is required to make the same choice in indistinguishable states. In Figure 10.3 the construction is shown for the formula $\varphi \equiv \exists x_1 \forall x_2 \exists x_3((x_1 \wedge x_2) \vee \neg x_3) \wedge (\neg x_1 \vee x_3)$. Finally, the model $\mathfrak{M}_\varphi$ is obtained from $\mathfrak{M}'_\varphi$ by following the same steps as described in Appendix B.3.

Secondly, we construct formula $\theta_\varphi$ from $\varphi$ as follows:

$$\theta_\varphi \equiv (\textbf{set-pl}\ \sigma_1.\forall \sigma_2 \exists \sigma_3 \ldots Q_n \sigma_n \chi)\mathbf{Pl}\,\langle\!\langle \mathbb{A}\mathrm{gt} \rangle\!\rangle \bigcirc \top$$

where

$$\chi \equiv \left( \bigwedge_{i=1,\ldots,n} \mathit{uniform}^i_{\mathsf{ATLP}}(\sigma_i) \right) \wedge (\textbf{set-pl}\ \langle \sigma_1[1], \ldots, \sigma_n[n] \rangle)\mathbf{Pl}\,\langle\!\langle \emptyset \rangle\!\rangle \Diamond \mathsf{yes}.$$

Next, we will give the intuition behind $\theta_\varphi$. Firstly, it is easy to see that $\mathbf{Pl}\,\langle\!\langle \mathbb{A}\mathrm{gt} \rangle\!\rangle \bigcirc \top$ is true whenever the set of plausible strategy profiles is *not empty*. Hence, the actual set of strategies described by the preceding (**set-pl** $\cdot$) operator is not particularly important, rather if *some* strategy is plausible or not.

Secondly, note that (**set-pl** $\langle \sigma_1[1], \ldots, \sigma_n[n] \rangle$) in $\chi$ describes a single strategy profile and that all individual strategies can be considered independently (the set is rectangular, cf. Definition 10.10 and Lemma 10.12). Furthermore, an individual strategy is mainly used to assign $\top$ or $\bot$ to propositional variables in the proposition states. (Except for agents $a_1$ and $a_2$ which also take on the refuter and verifier role; they can also perform actions in non-proposition states.) Hence, a given strategy profile can be seen as a valuation of the propositional variables.

Thirdly, we analyse $\chi$ with respect to a given profile $\sigma := \langle\sigma_1[1], \ldots, \sigma_n[n]\rangle$ taking into account the previous points. By formula $uniform^i_{\mathsf{ATLP}}(\sigma_i)$ it is ensured that agent $i$ assigns the same valuation to propositions in indistinguishable states. Now, $\chi$ is true if the "winning state" $q_\top$ is reached by following the strategy described by $\sigma$ (it describes a unique path in the model). In other words, $\chi$ is true if, and only if, the valuation described by $\sigma$ satisfies $\varphi$.

Finally, due to the previous observations, if $[\![\sigma_1.\forall\widehat{\sigma_2\exists\sigma_3\ldots} Q_n\sigma_n\chi]\!]$ is nonempty it can be interpreted as follows: There is a valuation of $x_1$ such that for all valuations of $x_2$ there is a valuation of $x_3$, and so forth such that $\varphi$ is satisfied.

The following proposition states that the construction is correct.

**Proposition 10.32.** *Let $\varphi$ be a* QSAT *instance. Then it holds that $\varphi$ is satisfiable if, and only if, $\mathfrak{M}_\varphi, q_0 \models \theta_\varphi$ where $\mathfrak{M}_\varphi$ and $\theta_\varphi$ are effectively constructible from $\varphi$ in polynomial time with respect to the length of the formula $\varphi$.*

The complete proof is given on page 312. We get the following theorem.

**Theorem 10.33 ($\mathcal{L}^k_{ATLP}$ is PSPACE-complete).** *The model checking problems for* **ATLP** *and for* $\mathbf{ATLP}^k$ *(for each $k \geq 1$) are* **PSPACE***-complete.*

*Proof.* Easiness is immediate since the model checking algorithm presented in Figure 10.6 can be executed in polynomial space with respect to the input (cf. Theorem 10.29 and Proposition 10.7). Hardness is shown by the polynomial space reduction from QSAT (Proposition 10.32). ■

Finally, we turn to classes in which the number of alternations is restricted by a fixed upper bound, and we conjecture that the model checking problem for $i$-level formulae of $\mathcal{L}^k_{ATLP}$ is in fact complete in its complexity classes determined in Theorems 10.29 and 10.30.

*Conjecture 10.34.* Let $\varphi$ be a level-$i$ formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \emptyset)$, $k \geq 1$, $i \geq 0$. Moreover, let $\mathfrak{M}$ be a CGS, and $q$ a state in $\mathfrak{M}$. Then, model checking $\mathfrak{M}, q \models \varphi$ is $\mathbf{\Delta^P_{i+2k+1}}$-complete.

*Conjecture 10.35.* Let $\varphi$ be a level-$i$ formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$, $\mathfrak{M}$ a well-behaved CGSP, and $q$ a state in $\mathfrak{M}$. Model checking $\mathfrak{M}, q \models \varphi$ is $\mathbf{\Delta^P_{i+2(k+1)+1}}$-complete.

| | 0 | 1 | 2 | ... | $i$ | ... | unbounded |
|---|---|---|---|---|---|---|---|
| $\mathcal{L}^{\text{basic}}_{ATLP}$ | $\mathbf{P}$ | - | - | - | - | ... | - |
| $\mathcal{L}^{0}_{ATLP}$ | $\mathbf{P}$ | - | - | ... | - | ... | - |
| $\mathcal{L}^{1}_{ATLP}$ | $\mathbf{\Delta^P_3}$ | $\mathbf{\Delta^P_4}$ | $\mathbf{\Delta^P_5}$ | ... | $\mathbf{\Delta^P_{i+3}}$ | ... | **PSPACE** |
| $\mathcal{L}^{2}_{ATLP}$ | $\mathbf{\Delta^P_5}$ | $\mathbf{\Delta^P_6}$ | $\mathbf{\Delta^P_7}$ | ... | $\mathbf{\Delta^P_{i+5}}$ | ... | **PSPACE** |
| $\vdots$ | | | | | | ... | $\vdots$ |
| $\mathcal{L}^{k}_{ATLP}$ $i > k+1$ | $\mathbf{\Delta^P_{2k+1}}$ | $\mathbf{\Delta^P_{2k+2}}$ | $\mathbf{\Delta^P_{2k+3}}$ | ... | $\mathbf{\Delta^P_{i+2k+1}}$ | ... | **PSPACE** |

**Fig. 10.4.** Summary of the model checking results for pure concurrent game structures (i.e., without hard-wired plausibility terms). All $\mathbf{P}$, $\mathbf{\Delta^P_3}$, and **PSPACE** results are completeness results.

### 10.1.6 Summary of these results

Throughout Section 10.1, we have analysed the model checking complexity of **ATLP**. The base language was shown to lie in $\mathbf{\Delta^P_3}$ with both abstract and $\mathcal{L}_{ATLI}$-based plausibility terms. We also proved that model checking both logics is complete regarding this class. The complexity of model checking **ATLP**$^k$ was shown to depend on three factors:

1. The *nesting level* $k$ of plausibility terms;
2. the *quantifier level*; and
3. whether abstract plausibility terms were present or not.

The quantifier level is influenced by the number of alternations and with which quantifiers – existential or universal – sequences start and end. In general, an $i$-level $\mathcal{L}^k_{ATLP}$-formula without plausibility terms was shown to be in

$$\mathbf{\Delta^P_{i+2k+1}}$$

where its counterpart with hard-wired terms was marginally more difficult to check:

$$\mathbf{\Delta^P_{i+2(k+1)+1}}.$$

The results for formulae without (resp. with) primitive plausibility terms are summarised in Figure 10.4 (resp. Figure 10.5).

Note that all our game theoretic characterisations could already be expressed by $\mathcal{L}^1_{ATLP}$-formulae without hard-wired terms.

| | 0 | 1 | 2 | ... | $i$ | ... | unbounded |
|---|---|---|---|---|---|---|---|
| $\mathcal{L}^{\text{basic}}_{ATLP}$ | $\mathbf{\Delta^P_3}$ | - | - | ... | - | ... | - |
| $\mathcal{L}^{0}_{ATLP}$ | $\mathbf{\Delta^P_3}$ | - | - | ... | - | ... | - |
| $\mathcal{L}^{1}_{ATLP}$ | $\mathbf{\Delta^P_5}$ | $\mathbf{\Delta^P_6}$ | $\mathbf{\Delta^P_7}$ | ... | $\mathbf{\Delta^P_{i+5}}$ | ... | **PSPACE** |
| $\mathcal{L}^{2}_{ATLP}$ | $\mathbf{\Delta^P_7}$ | $\mathbf{\Delta^P_8}$ | $\mathbf{\Delta^P_9}$ | ... | $\mathbf{\Delta^P_{i+7}}$ | ... | **PSPACE** |
| ⋮ | | | | | | | ⋮ |
| $\mathcal{L}^{k}_{ATLP}$ $i > k$ | $\mathbf{\Delta^P_{2k+3}}$ | $\mathbf{\Delta^P_{2k+4}}$ | $\mathbf{\Delta^P_{2k+5}}$ | ... | $\mathbf{\Delta^P_{i+2(k+1)+1}}$ | ... | **PSPACE** |

**Fig. 10.5.** Summary of the model checking results in well-behaved CGSPs. All $\mathbf{\Delta^P_3}$ and **PSPACE** results are completeness results.

## 10.2 Abilities of Rational Coalition: **CoalATL**

In this section we present an algorithm for model checking **CoalATL**.

### 10.2.1 Easiness

For **CoalATL** we also have to treat the new coalitional modalities in addition to the normal **ATL** constructs. Let us consider the formula $\langle\!| A |\!\rangle\psi$. According to the semantics of $\langle\!| A |\!\rangle$, given in Definition 6.50, we must check whether there is a coalition $B$ such that (i) if $A \neq \emptyset$ then $A \cap B \neq \emptyset$, (ii) $B$ is acceptable by the argumentation semantics, and (iii) $\langle\!\langle B \rangle\!\rangle\psi$. The number of possible candidate coalitions $B$ which satisfy (i) and (ii) is bounded by $|\mathcal{P}(\mathbb{A}\mathrm{gt})|$. Thus, in the worst case there might be *exponentially* many calls to a procedure checking whether $\langle\!\langle B \rangle\!\rangle\psi$. Another source of complexity is the time needed to compute the argumentation semantics.

Both considerations together suggest that the model checking complexity has two computationally hard parts: exponentially many calls to $\langle\!\langle B \rangle\!\rangle\psi$ and the computation of the argumentation semantics. Indeed, Theorem 10.37 will support this intuition. However, we show that it is possible to "combine" both computationally hard parts to obtain an algorithm which is in $\mathbf{\Delta^P_2} = \mathbf{P^{NP}}$, if the computational complexity to determine whether *a given* coalition is acceptable is in **NP**.

For the rest of this section, we will denote by $\mathcal{VER}_{\mathfrak{sem}}(\mathcal{CF}, A)$ the *verification problem* (cf. [Dunne and Caminada, 2008]) which represents the problem whether for a given argumentation semantics $\mathfrak{sem}$, a coalitional framework $\mathcal{CF}$, and coalition $A \subseteq \mathbb{A}\mathrm{gt}$ we have that $A \in \mathfrak{sem}(\mathcal{CF})$. Given some complexity class $\mathcal{C}$, we use the notation "$\mathcal{VER}_{\mathfrak{sem}} \in \mathcal{C}$" to state that the verification problem with respect to the semantics $\mathfrak{sem}$ is in $\mathcal{C}$.

**function** *mcheck*$(\mathfrak{M}, q, \varphi)$**;**

Given a CGM $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \zeta, \mathfrak{sem}\rangle$, a state $q \in Q$, and $\varphi \in \mathcal{L}_[ATL^c](\mathbb{A}\mathrm{gt}, \Pi)$ the algorithm returns $\top$ if, and only if, $\mathfrak{M}, q \models_{\mathfrak{sem}} \varphi$.

case $\varphi$ contains no $\langle\!|B|\!\rangle$: **if** $q \in mcheck_{\mathsf{ATL}}(\mathfrak{M}, \varphi)$ **return** $\top$ **else** $\bot$
case $\varphi$ contains some $\langle\!|B|\!\rangle$:
- case $\varphi \equiv \neg\psi$: **return** $\neg(\mathfrak{M}, q, \psi)$
- case $\varphi \equiv \psi \vee \psi'$: **return** $mcheck(\mathfrak{M}, q, \psi) \vee mcheck(\mathfrak{M}, q, \psi')$
- case $\varphi \equiv \langle\!\langle A\rangle\!\rangle T\psi$: Label all states $q'$ where $mcheck(\mathfrak{M}, q', \psi) == \top$ with a new proposition $\mathsf{yes}$ and return $mcheck(\mathfrak{M}, q, \langle\!\langle A\rangle\!\rangle T\mathsf{yes})$; $T$ stands for $\Box$ or $\bigcirc$.
- case $\varphi \equiv \langle\!\langle A\rangle\!\rangle \psi\mathcal{U}\psi'$: Label all states $q'$ where $mcheck(\mathfrak{M}, q', \psi) == \top$ with a new proposition $\mathsf{yes}_1$, all states $q'$ where $mcheck(\mathfrak{M}, q', \psi') == \top$ with a new proposition $\mathsf{yes}_2$ and return $mcheck(\mathfrak{M}, q, \langle\!|A|\!\rangle \mathsf{yes}_1\mathcal{U}\mathsf{yes}_2)$
- case $\varphi \equiv \langle\!|A|\!\rangle T\psi$, $\psi$ contains some $\langle\!|C|\!\rangle$: Label all states $q'$ where $mcheck(\mathfrak{M}, q', \psi) == \top$ with a new proposition $\mathsf{yes}$ and return $mcheck(\mathfrak{M}, q, \langle\!|A|\!\rangle T\mathsf{yes})$; $T$ stands for $\Box$ or $\bigcirc$.
- case $\varphi \equiv \langle\!|A|\!\rangle \psi\mathcal{U}\psi'$, $\psi$ or $\psi'$ contain some $\langle\!|C|\!\rangle$: Label all states $q'$ where $mcheck(\mathfrak{M}, q', \psi) == \top$ with a new proposition $\mathsf{yes}_1$, all states $q'$ where $mcheck(\mathfrak{M}, q', \psi') == \top$ with a new proposition $\mathsf{yes}_2$ and return $mcheck(\mathfrak{M}, q, \langle\!|A|\!\rangle \mathsf{yes}_1\mathcal{U}\mathsf{yes}_2)$
- case $\varphi \equiv \langle\!|A|\!\rangle\psi$ and $\psi$ contains no $\langle\!|C|\!\rangle$: Non-deterministically choose $B \in \mathcal{P}(\mathbb{A}\mathrm{gt})$
  **if**
  (1) $B \in (\mathfrak{sem}(\zeta(A)(q)))$,
  (2) if $A \neq \emptyset$ then $A \cap B \neq \emptyset$, and $(\star)$
  (3) $q \in mcheck_{\mathsf{ATL}}(\mathfrak{M}, \langle\!\langle B\rangle\!\rangle\psi)$
  **then return** $\top$ **else** $\bot$

**function** $mcheck_{\mathbf{ATL}}(\mathfrak{M}, \varphi)$**;**

Given a CGS $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o\rangle$ and $\varphi \in \mathcal{L}_{ATL}(\mathbb{A}\mathrm{gt}, \Pi)$, the standard **ATL** model checking algorithm (cf. [Alur et al., 2002]) returns all states $q$ with $\mathfrak{M}, q \models_{\mathsf{ATL}} \varphi$.

- return $\{q \in Q \mid \mathfrak{M}, q \models_{\mathsf{ATL}} \varphi\}$

**Fig. 10.6.** A model checking algorithm for **CoalATL**.

In [Bulling et al., 2008] it is stated that $\mathcal{VER}_{\mathfrak{sem}} \in \mathbf{P}$ for all semantics introduced in Definition 4.50. Unfortunately, this result is incorrect for the preferred semantics. There is a flaw in the way maximal sets of coalitions are treated. Actually, from [Dung, 1995] it follows that $\mathcal{VER}_{\mathfrak{sem}} \in \mathbf{P}$ for $\mathfrak{sem} \in \{\mathfrak{sem}_{\text{admissible}}, \mathfrak{sem}_{\text{grounded}}, \mathfrak{sem}_{\text{stable}}\}$ and in [Dimopoulos and Torres, 1996] it was shown that $\mathcal{VER}_{\mathfrak{sem}_{\text{preferred}}} \in \mathbf{coNP}$-complete. In the following proposition, we summarise these results and also treat the complete semantics. For an overview we refer to [Dunne and Caminada, 2008].

**Proposition 10.36 ([Dung, 1995; Dimopoulos and Torres, 1996]).** *We have that* $\mathcal{VER}_{\mathfrak{sem}} \in \mathbf{P}$ *for all semantics* $\mathfrak{sem} \in \{\mathfrak{sem}_{admissible}, \mathfrak{sem}_{grounded}, \mathfrak{sem}_{stable}, \mathfrak{sem}_{complete}\}$ *and* $\mathcal{VER}_{\mathfrak{sem}_{preferred}} \in \mathbf{coNP}$*-complete.*

*Proof.* It remains to show the case for $\mathcal{VER}_{\mathfrak{sem}_{\text{complete}}}$. Note that computing $\mathcal{F}_{\mathcal{CF}}(\mathfrak{S})$ for a given $\mathfrak{S}$ can be done in polynomial time. Therefore, to check whether a set of coalitions $\mathfrak{S}$ is complete, we can check that it is admissible and that all elements of $\mathcal{F}_{\mathcal{CF}}(\mathfrak{S})$ are already contained in $\mathfrak{S}$. Both checks can be done in polynomial time. Thus $\mathcal{VER}_{\mathfrak{sem}_{\text{complete}}} \in \mathbf{P}$. ■

In Figure 10.6 we propose a model checking algorithm for **CoalATL**. The complexity result given in the next theorem is modulo the complexity needed to solve the verification problem $\mathcal{VER}_{\mathfrak{sem}}$.

**Theorem 10.37 (Model checking CoalATL).** *Let a* CGM

$$\mathfrak{M} = \langle \mathbb{A}\text{gt}, Q, \Pi, \pi, Act, d, o, \zeta, \mathfrak{sem} \rangle$$

*be given,* $q \in Q$*,* $\varphi \in \mathcal{L}_{CoalATL}(\mathbb{A}\text{gt}, \Pi)$*, and* $\mathcal{VER}_{\mathfrak{sem}} \in \mathcal{C}$*. Model checking* **CoalATL** *with respect to the argumentation semantics* $\mathfrak{sem}$[4] *is in* $\mathbf{P}^{\mathbf{NP}^{\mathcal{C}}}$*.*

*Proof.* The algorithm $mcheck_{\mathsf{ATL}}$ is $\mathbf{P}$-complete [Alur et al., 2002]. The number of modalities $\langle\!| \cdot |\!\rangle$ is bounded by $|\varphi|$. Thus, the last case can only be performed a polynomial number of times (wrt. the length of $\varphi$). The complexity of the last case is as follows. Firstly, $B$ is guessed and then verified. The verification is performed by an oracle call (with complexity $\mathcal{C}$) to check whether $B \in \mathfrak{sem}(\zeta(A)(q))$ and two additional steps which can be performed by a deterministic Turing machine in polynomial time. ■

The last theorem gives an upper bound for model checking **CoalATL** with respect to an arbitrary but fixed semantics $\mathfrak{sem}$. A finer grained classification of the computational complexity of $\mathcal{VER}_{\mathfrak{sem}}$ allows to improve the upper bound given in Theorem 10.37. Assume that $\mathcal{VER}_{\mathfrak{sem}} \in \mathbf{NP}$; then, a witness can be non-deterministically guessed together with the coalition $B \in \mathcal{P}(\mathbb{A}\text{gt})$ and then, it is checked whether $B$ satisfies the three conditions (1-3) in ($\star$). Each of the three cases can be done in deterministic polynomial time. Hence, the verification of $\mathfrak{M}, q \models \langle\!| A |\!\rangle \psi$, in the last case, meets the "guess and verify" principle which is characteristic for problems in $\mathbf{NP}$. This brings the overall complexity of the algorithm to $\mathbf{\Delta_2^P}$.

---

[4] That is, whether $\mathfrak{M}, q \models_{\mathfrak{sem}} \varphi$.

**Corollary 10.38 ([Bulling et al., 2008]).** *If $\mathcal{VER}_{\mathfrak{sem}} \in \mathbf{NP}$ then model checking* **CoalATL** *is in $\mathbf{\Delta_2^P}$ with respect to $\mathfrak{sem}$.*

*Proof.* Since $\mathcal{VER}_{\mathfrak{sem}}$ is in **NP** there is a deterministic Turing machine $\mathfrak{M}$ that runs in polynomial time $p(n)$ (where $n$ is the length of the input $(A, \mathcal{CF})$) that accepts $((A, \mathcal{CF}), w)$ for some witness $w$ of length less or equal $p(n)$ iff $A \in \mathfrak{sem}(\mathcal{CF})$ (otherwise it does not accept $((A, \mathcal{CF}), w)$ for all witnesses $w$ with $|w| \leq p(n)$). Now, in the model checking algorithm, we extend the non-deterministic guess of the coalition $B$ in $(\star)$ by also guessing a witness $w$ for the input of machine $\mathfrak{M}$. This can be implemented by a single non-deterministic machine (e.g., using the deterministic machine $\mathfrak{M}$ as oracle or implementing it directly). Then, the whole algorithm is in $\mathbf{P}^{\mathbf{NP}^{\mathbf{P}}} = \mathbf{P}^{\mathbf{NP}} = \mathbf{\Delta_2^P}$. ■

In the line with Proposition 10.36 we modify the result from [Bulling et al., 2008] as follows.

**Corollary 10.39.** *Model checking* **CoalATL** *is in $\mathbf{\Delta_2^P}$ for $\mathfrak{sem}_{admissible}$, $\mathfrak{sem}_{complete}$ and $\mathfrak{sem}_{stable}$.*

The following result is immediate as $\mathcal{VER}_{\mathfrak{sem}_{\text{preferred}}}$ is **coNP**-complete.

**Corollary 10.40.** *Model checking* **CoalATL** *is in $\mathbf{\Delta_3^P}$ for $\mathfrak{sem}_{preferred}$.*

As the next proposition shows, the model checking algorithm can also be improved in the cases that only polynomially many coalitions are acceptable wrt the semantics and that all these coalitions can be computed in polynomial time.

**Proposition 10.41.** *Model checking* **CoalATL** *is* **P***-complete for semantics $\mathfrak{sem}$ that only accept polynomially many coalitions and for which it is possible to enumerate all theses coalitions in polynomial time with respect to the size of the model and the length of the formula.*

*Proof.* For a formula $\langle\!| A |\!\rangle\gamma$ we verify whether $\langle\!\langle B \rangle\!\rangle\gamma$ for all the polynomially many coalitions $B$ acceptable by $\mathfrak{sem}$. Completeness follows from the completeness of model checking pure **ATL** [Alur et al., 2002]. ■

Since the grounded semantics is characterised by the smallest fixed point there only is a unique coalition. Moreover, the fixed point can be calculated on polynomial time. So, the following result is immediate.

**Corollary 10.42.** *Model checking* **CoalATL** *is* **P***-complete for $\mathfrak{sem}_{grounded}$ (and thus also for $\mathfrak{sem}_{cs}$) .*

### 10.2.2 Definitions and NP/coNP-hardness

In the next section we show that model checking **CoalATL** is $\mathbf{\Delta_2^P}$-hard even for the very simple argumentation semantics that can essentially characterise *all* truth assignments of Boolean formulae. In this section, we firstly show that the model checking problem is **NP**-hard and **coNP**-hard by reducing SAT [Papadimitriou, 1994] (satisfiability of Boolean formulae (in positive normal form)) to model checking **CoalATL** and introduce the basic definitions needed for the $\mathbf{\Delta_2^P}$-hardness proof presented in the following section.

Let $\varphi = \varphi(X)$ be a Boolean formula in positive normal form[5] over the Boolean variables $X := \{x_1, \ldots, x_n\}$. A *truth assignment* of a Boolean formula $\varphi(X)$ is a mapping $X \to \{0, 1\}$. We identify a truth assignment with the set $X' \subseteq X$ of variables that are assigned 1 (true). We define a necessary condition on the expressiveness of argumentation semantics which forces the problem to become complete.

**Definition 10.43 (Reduction-suitable semantics, $\mathfrak{sem}$-witness).** *Let $X_1$ and $X_2$ be two non-empty and disjoint sets of the same size and $f : X_1 \to X_2$ a bijective mapping between these sets and let $x$ be an element not in $X_1 \cup X_2$. Moreover, let $X_1 \cup X_2 \cup \{x\} \subseteq Y$ for some set $Y$.*

*We call a semantics $\mathfrak{sem}$ over $Y$* reduction-suitable *if for any sets $X_1, X_2$, $\{x\}$ satisfying the properties given above there is a coalitional framework whose size is polynomial in $Y$ such that $E \in \mathfrak{sem}(\mathcal{CF})$ iff (1) $E = X \cup \{x\}$, (2) $X \subseteq X_1 \cup X_2$ and (3) $\forall x \in X_1 \cup X_2$ ($x \in X$ iff $f(x) \notin X$).*

*Moreover, we call a coalitional framework that witnesses that a semantics is reduction-suitable a* $\mathfrak{sem}$-witness *coalitional framework.*

A reduction-suitable semantics allows to describe all truth assignments of a formula in a compact way. The intuition is that $X_1$ and $X_2$ represent the variables and their negations, respectively. The set $X$ with $X \cup \{x\} \in \mathfrak{sem}(\mathcal{CF})$ encodes a truth assignment; i.e. the *literals* assigned 1. The variable $x$ is a technicality needed in the reduction. In the following proposition we make the observation that reduction-suitable semantics allow to represent exactly all truth assignments.

**Proposition 10.44.** *Let $\varphi = \varphi(X)$ be a Boolean formula and let $\bar{X} := \{\bar{x} \mid x \in X\}$ and let $\mathfrak{sem}$ be a reduction-suitable semantics over $Y$ where $X \cup \bar{X} \cup \{x\} \subseteq Y$ and $x \notin X$. Then, there is a coalitional framework $\mathcal{CF}$ such that for each $T \in \mathfrak{sem}(\mathcal{CF})$, $T \cap X$ is a truth assignment of $\varphi$ and for each truth assignment $T$ of $\varphi$ there is a $Z \in \mathfrak{sem}(\mathcal{CF})$ such that $Z \cap X = T$.*

---

[5] That is, negation symbols do only occur at variables. Note that the positive normal form can be established in polynomial time.

*Proof.* Let $\mathcal{CF}$ be a $\mathfrak{sem}$-witness. Firstly, let $T \in \mathfrak{sem}(\mathcal{CF})$, clearly $T \cap X$ does only contain elements of $X$ and thus is a truth assignment. Now let, $T$ be a truth assignment of $\varphi$. We show that $Z \cup \{x\} \in \mathfrak{sem}(\mathcal{CF})$ with $Z := T \cup X'$, $X' := \{\bar{x} \mid x \notin T\}$ and $(Z \cup \{x\}) \cap X = T$. Clearly, we have that $Z \subseteq X \cup \bar{X}$ and also that ($x \in Z$ iff $\bar{x} \notin Z$) for all $\forall x \in X \cup \bar{X}$. This shows that both conditions of reduction-suitable semantics are satisfied. ■

In the following we introduce some notation to refer to subformulae of a formula and to the outermost logical Boolean connector $\vee$ or $\wedge$.

**Definition 10.45 (Notation for subformulae of $\varphi$, relevance).** *Let $\varphi$ be a Boolean formula in positive normalform. We define $lc(\varphi)$ as the* outermost logical connector *in $\varphi$; that is,*

$$lc(\varphi) := \begin{cases} \wedge & \text{if } \varphi = \psi_1 \wedge \psi_2, \\ \vee & \text{if } \varphi = \psi_1 \vee \psi_2, \\ \epsilon & \text{if } \varphi \text{ is a literal.} \end{cases}$$

*Moreover, we define $ls(\varphi)$ (resp. $rs(\varphi)$) as the* subformula on the left hand side *(resp.* right hand side*) of $lc(\varphi)$ provided that $lc(\varphi) \neq \epsilon$. In the case of $lc(\varphi) = \epsilon$ we set $ls(\varphi) = rs(\varphi) = \varphi$. Note that we have that $\varphi = ls(\varphi)lc(\varphi)rs(\varphi)$ whenever $lc(\varphi) \neq \epsilon$.*

*Now, we can assign a string over $\{1,2\}$ to refer to a subformula of $\varphi$, where 1 (resp. 2) stands for the left (resp. right) subformula wrt to the outermost logical connector. Formally, we define a function $\chi^\varphi$ from $\{1,2\}^+ \cup \{0\}$ into the subformulae of $\varphi$ as follows (we write $\chi^\varphi_w$ for $\chi^\varphi(w)$ where $w \in \{1,2\}^+ \cup \{0\}$):*

$$\chi^\varphi_w := \begin{cases} \varphi & \text{if } w = 0, \\ ls(\varphi) & \text{if } w = 1, \\ rs(\varphi) & \text{if } w = 2, \\ ls(\chi^\varphi_x) & \text{if } w = x1,\ x \in \{1,2\}^+, \\ rs(\chi^\varphi_x) & \text{if } w = x2,\ x \in \{1,2\}^+. \end{cases}$$

*Finally, we call a string $w \in \{1,2\}^+$* relevant for $\varphi$ *iff $lc(\chi^\varphi_x) \neq \epsilon$ for $x \in \{1,2\}^*$ and $w = xi$ with $i \in \{1,2\}$; or if $w = 0$. We will also just write $\chi_w$ if the formula $\varphi$ is clear from context.*

Given the formula $\varphi = ((x_1 \wedge x_2) \vee \neg x_3) \wedge (\neg x_1 \vee x_3)$, for instance, we have that $\chi^\varphi_2 = \neg x_1 \vee x_3$, $\chi^\varphi_{112} = x_2$, and $\chi^\varphi_{21} = \neg x_1$,

We proceed with our reduction. Inspired by [Bulling et al., 2009b; Jamroga and Dix, 2008] we construct a CGM corresponding to $\varphi(X)$ which essentially corresponds to the parse tree of $\varphi(X)$ and implements the game semantics of Boolean formulae (cf. [Hintikka and Sandu, 1997]).

That is, we construct the CGM $\mathfrak{M}(\varphi)$ corresponding to $\varphi(X)$ with $2+2|X|$ players: *verifier* $\mathbf{v}$, *refuter* $\mathbf{r}$, and agents $a_i$ and $\bar{a}_i$ for each variable $x_i \in X$. The CGM is turn-based, that is, every state is "governed" by a single player who determines the next transition. Each subformula $\chi_{i_1 \dots i_l}$ of $\varphi$ has a corresponding state $q_{i_1 \dots i_l}$ in $\mathfrak{M}(\varphi)$ for $i_k \in \{1,2\}$, and $\chi_0$ the state $q_0$. If the outermost logical connective of $\varphi$ is $\wedge$, i.e. $lc(\varphi) = \wedge$ , the refuter decides at $q_0$ which subformula $\chi_i$ of $\varphi$ is to be satisfied (i.e. whether $\chi_1$ or $\chi_2$), by proceeding to the "subformula" state $q_i$ corresponding to $\chi_i$. If the outermost connective is $\vee$, the verifier decides which subformula $\chi_i$ of $\varphi$ will be attempted at $q_0$. This procedure is repeated until all subformulae are single literals. In the following we refer to the states corresponding to literals as *literal states*.

The difference from the construction from [Jamroga and Dix, 2008] is that formulae are in positive normal form (rather than CNF) and from [Jamroga and Dix, 2008; Bulling et al., 2009b] in the way in which literal states are treated: Literal states are governed by agents $a_i$ or $\bar{a}_i$. The values of the underlying propositional variables $x, y$ are declared at the literal states, and the outcome is computed. That is, if $a_j$ executes $\top$ for a positive literal, i.e. $\chi_{i_1 \dots i_l} = x_j$, at $q_{i_1 \dots i_l}$, then the system proceeds to the "winning" state $q_\top$; otherwise, the system goes to the "sink" state $q_\perp$. Analogously, if $\bar{a}_j$ executes $\perp$ for a negative literal, i.e. $\chi_{i_1 \dots i_l} = \neg x_j$, at $q_{i_1 \dots i_l}$, then the system proceeds to the "winning" state $q_\top$; otherwise, the system goes to the "sink" state $q_\perp$.

Finally, the idea is to use $\mathfrak{sem}$-witness coalitional frameworks (for some reduction-suitable semantics) to represent all valuations of $\varphi(X)$ such that there is a "successful coalition" among these coalitions (representing the valuations of $\varphi$) iff $\varphi(X)$ is satisfiable. An example of the construction is shown in Figure 10.7. Formally, the model is defined as follows.

**Definition 10.46 ($\mathfrak{M}(\varphi)$).** *Let $\varphi(X)$ be given. The model*

$$\mathfrak{M}(\varphi) = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \zeta, \mathfrak{sem} \rangle$$

*is defined as follows:*

- $\mathbb{A}\mathrm{gt} := \{\mathbf{v}, \mathbf{r}\} \cup \{a_i, \bar{a}_i \mid x_i \in X\}$
- $Q := \{q_0, q_\top, q_\perp\} \cup \{q_w \mid w \in \{1,2\}^+$ *relevant for* $\varphi\}$
- $\Pi := \{\mathsf{sat}\}$
- $\pi(q_\top) = \{\mathsf{sat}\}$
- $Act := \{1, 2, \top, \perp\}$
- $d_{\mathbf{r}}(q_w) = \{1,2\}$ *for each* $w$ *relevant for* $\varphi$ *with* $lc(\chi_w^\varphi) = \wedge$; $d_{\mathbf{v}}(q_w) = \{1,2\}$ *for each* $w$ *relevant for* $\varphi$ *with* $lc(\chi_w^\varphi) = \vee$; $d_{a_i}(q_w) = \{\top, \perp\}$ *for each* $w$ *relevant for* $\varphi$ *with* $\chi_w^\varphi = x_i$; $d_{\bar{a}_i}(q_w) = \{\top, \perp\}$ *for each* $w$ *relevant for* $\varphi$ *with* $\chi_w^\varphi = \neg x_i$; $d_x(q) = \{\top\}$ *for* $q \in \{q_\top, q_\perp\}$ *and* $x \in \mathbb{A}\mathrm{gt}$.
- *We note that the model is turn-based (except for the states* $q_\top$ *and* $q_\perp$*); that is, in each state only one agent can execute actions. Hence, transitions*

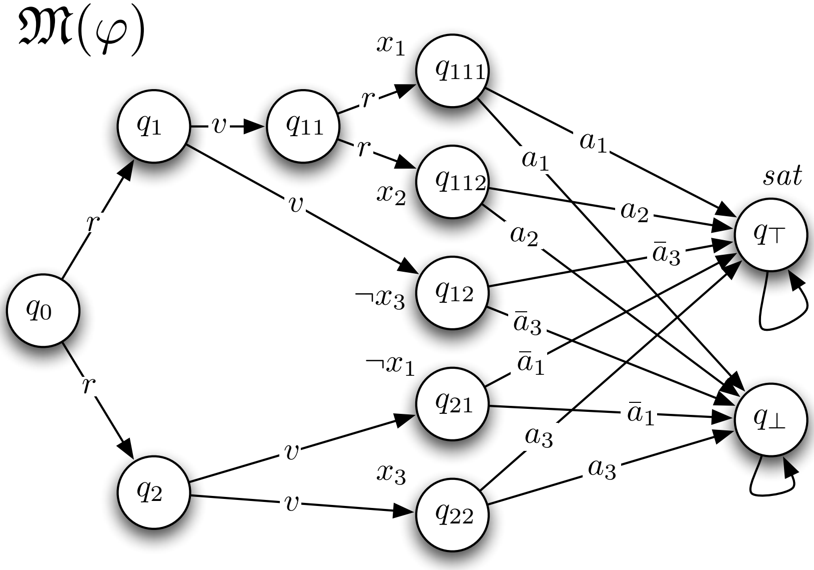



**Fig. 10.7.** The construction of $\mathfrak{M}(\varphi)$ for $\varphi = ((x_1 \wedge x_2) \vee \neg x_3) \wedge (\neg x_1 \vee x_3)$. Transitions labeled with an agent indicate that this agent can force the transition disregarding the behaviour of the other agents.

*do only depend on single actions and not on action profiles:* $o(q_0)(1) = q_1$, $o(q_0)(2) = q_2$, $o(q_w)(1) = q_{w1}$ *if* $w1$ *is relevant for* $\varphi$, $o(q_w)(2) = q_{w2}$ *if* $w2$ *is relevant for* $\varphi$; $o(q_w)(\top) = q_\top$ *(resp.* $o(q_w)(\bot) = q_\bot$*) if* $\chi_w^\varphi \in X$; $o(q_w)(\top) = q_\bot$ *(resp.* $o(q_w)(\bot) = q_\top$*) if* $\neg\chi_w^\varphi \in X$ *(we identify* $x$ *with* $\neg\neg x$*); and* $o(q)(\top) = q$ *for* $q \in \{q_\top, q_\bot\}$.
- $\mathfrak{sem}$ *is a reduction-suitable semantics over* $\mathbb{A}\mathrm{gt}$*. Finally, we set* $\zeta(A)(q) = \mathcal{CF}$ *where* $\mathcal{CF}$ *is some* $\mathfrak{sem}$*-witness coalitional framework for the sets* $X_1 := \{a_i \mid x_i \in X\}$, $X_2 := \{\bar{a}_i \mid x_i \in X\}$*, and* $x := \{\mathbf{v}\}$ *for all* $A \subseteq \mathbb{A}\mathrm{gt}$ *and* $q \in Q$.

We define a **v**-*choice of* $\mathfrak{M}(\varphi)$ as "the graph" that occurs if from states controlled by the verifier **v** all transitions but one are removed (i.e. at each state controlled by the verifier it does only have one action to execute). In other words, we fix a strategy of **v**. The following lemma is essential for our reduction.

**Lemma 10.47.** *Let* $\varphi(X)$ *be a Boolean formula in positive normal form.*

*(a) If $T$ is a satisfying truth assignment of $\varphi$, then there is a* $\mathbf{v}$*-choice of* $\mathfrak{M}(\varphi)$ *such that for the set $L$ of all literal states reachable from $q_0$ it holds that $\{x \in X \mid q_w \in L \text{ and } \chi_w = x\} \subseteq T$ and $\{x \in X \mid q_w \in L \text{ and } \chi_w = \neg x\} \cap T = \emptyset$.*

*(b) If there is a* $\mathbf{v}$*-choice of* $\mathfrak{M}(\varphi)$ *such that for the set $L$ of all literal states reachable from $q_0$ we have that for any $q_w, q_v \in L$ the formula $\chi_w \wedge \chi_v$ is satisfiable (i.e. there are no complementary literals) then the set $\{x \in X \mid q_w \in L \text{ and } \chi_w = x\}$ is a satisfying truth assignment of $\varphi$.*

The complete proof is given on page 316.

Similar to [Bulling et al., 2009b; Jamroga and Dix, 2008], we have the following result which shows that the construction above is a polynomial time reduction of SAT to model checking **CoalATL**.

**Proposition 10.48.** *The model $\mathfrak{M}(\varphi)$ is constructible in polynomial-time wrt the size of $\varphi$ and we have that*

$$\varphi(X) \text{ is satisfiable if, and only if, } \mathfrak{M}(\varphi), q_0 \models \langle\!\langle \mathbf{v} \rangle\!\rangle \Diamond \mathsf{sat}.$$

The complete proof is given on page 318.

The following result is obvious, we can reduce SAT and UNSAT to model checking $\mathfrak{M}(\varphi), q_0 \models \langle\!\langle \mathbf{v} \rangle\!\rangle \Diamond \mathsf{sat}$ and $\mathfrak{M}(\varphi), q_0 \models \neg\langle\!\langle \mathbf{v} \rangle\!\rangle \Diamond \mathsf{sat}$, respectively.

**Theorem 10.49.** *Model checking* **CoalATL** *is* **NP***-hard and* **coNP***-hard for any reduction-suitable semantics.*

### 10.2.3 $\mathbf{\Delta_2^P}$-hardness

Finally, we show our main result, the $\mathbf{\Delta_2^P}$-hardness of model checking **CoalATL** for reduction-suitable semantics. We do so by reducing $\text{SNSAT}_1$, a typical $\mathbf{\Delta_2^P}$-complete problem stated in Definition 4.15.

For this section, we also use the following notation for an $\text{SNSAT}_1$-instance:

$$z^r \equiv \exists Y_r(\varphi_r(z^1, ..., z^{r-1}, X^r))$$

where $Y_r \subseteq X^r$. The notation is understood as follows: There is a truth assignment assigning 1 (resp. 0) to the variables in $Y_r$ (resp. $X^r \backslash Y_r$) such that $\varphi_r(z^1, ..., z^{r-1}, X^r)$ is true under this assignment. We use $I = (\varphi_1(X^1), \ldots, \varphi_p(X^p))$ or just $I = (\varphi_1, \ldots, \varphi_p)$ to denote an *instance* of $\text{SNSAT}_1$ and set $Z = \{z^1, \ldots, z^p\}$.

We often need to analyse a solution of an $\text{SNSAT}_1$-instance. In the following we show how a solution can formally be stated.

**Definition 10.50 (Witness and solution of an $\text{SNSAT}_1$-instance).** *Let $I = (\varphi_1, \ldots, \varphi_p)$ be an $\text{SNSAT}_1$-instance. A tuple $(T_1, \ldots, T_p)$ is an $I$-*witness *if it satisfies the following properties:*

1. *$T_i \subseteq \{z^1, \ldots, z^i\} \cup X^i$;*
2. *If $\varphi_i$ is satisfiable under the partial assignment $\{z^j \in T_j \mid j < i\}$ then $T_i$ is a satisfying truth assignment of $\varphi_i$ and $z^i \in T_i$; else $z^i \notin T_i$ and $T_i \cap X^i = \emptyset$;*
3. *$z^i \in T_i$ implies $z^i \in T_j$ for all $j \geq i$,*

*An $I$-witness $T$ is a* solution of $I$ *iff $z^p \in T_p$.*

In the next proposition we state some properties about solutions and witnesses.

**Proposition 10.51.** *Let $I = (\varphi_1, \ldots, \varphi_p)$ be an* $\text{SNSAT}_1$ *instance and let $T = (T_1, \ldots, T_p)$ be an $I$-witness.*

*(a) Let $T' = (T'_1, \ldots, T'_p)$ be another $I$-witness; then, $T'_i \cap Z = T_i \cap Z$ for all $i = 1, \ldots, p$. (That is, among all witnesses the values for the $z^i$'s are uniquely determined.)*
*(b) If $I^i = (\varphi_1, \ldots, \varphi_i)$ with $i < p$ has a solution, then $T = (T_1, \ldots, T_i)$ is a solution of $I^i$.*
*(c) Let $I' = (\varphi_1, \ldots, \varphi_p, \varphi_{p+1})$ be such that $I'$ has no solution; then, $T' = (T_1, \ldots, T_p, \{z \in T_p\})$ is an $I'$-witness.*

*Proof.*

(a) Suppose that $i$ is the minimal index for which both sets $T_i \cap Z$ and $T'_i \cap Z$ differ; that is, wlog, $z_i \in T_i$ and $z_i \notin T'_i$. This is a contradiction to property (2) of the definition of a witness; as the satisfiability of $\varphi_i$ under $z_1, \ldots z_{i-1}$ is uniquely determined.
(b) Follows from (a).

(c) We show that $R := \{z \in T_p\}$ satisfies the conditions (1-3) in the Definition of a witness. Clearly, $R \subseteq \{z^1, \ldots, z^{p+1}\}$. Since $I'$ has no solution, there is no way to make $z_{p+1}$ true. Clearly, $z_{p+1} \notin R$ and $R \cap X^{p+1} = \emptyset$. Finally, condition (3) is satisfied by definition of $R$.

■

Our reduction of $\text{SNSAT}_1$ is a modification of the reduction of $\text{SNSAT}_2$ presented in [Bulling et al., 2009b; Jamroga and Dix, 2008] and extends the **NP/coNP**-hardness construction of the previous section. Consider an $\text{SNSAT}_1$ instance $I = (\varphi_1, \ldots, \varphi_p)$. Essentially, we construct models $\mathfrak{M}(\varphi_r)$ for $r = 1, \ldots, p$ as shown above but we label each state of $\mathfrak{M}(\varphi_r)$ and each agent name by an additional superindex $r$ (that is, states are denoted by $q_w^r$ and agents by $a_i^r$ and $\bar{a}_i^r$). The main difference is how the literal states corresponding to literals $z_i$ and $\neg z_i$ are treated. We connect such states of model $\mathfrak{M}(\varphi_r)$ with $r > 1$ with the initial state $q_0^{r-1}$ of model $\mathfrak{M}(\varphi_{r-1})$. Additionally, states referring to negated variables $z_i$ are labeled with a proposition $\mathsf{neg}$. Finally, the full model $\mathfrak{M}(I)$ wrt an $\text{SNSAT}_1$ instance $I$ is given by the combination of these model as just explained. In particular, the set of agents is given by $\{\mathbf{v}, \mathbf{r}\} \cup \{a_i^r, \bar{a}_i^r \mid x_i^r \in \bigcup_{j=1}^{p} X^j\}$. An example of the construction is shown in Figure 10.8. Given the model $\mathfrak{M}(I)$ we use $\mathfrak{M}(\varphi_i)$ to refer the restriction of $\mathfrak{M}(I)$ to the states with superindex $i$. In Figure 10.8 the two submodels $\mathfrak{M}(\varphi_1)$ and $\mathfrak{M}(\varphi_2)$ are framed.

The formulae used in this reduction are more sophisticated as they have to account for the nested structure of an $\text{SNSAT}_1$ instance. We define

$$\varphi_0 \equiv \top$$

and

$$\varphi_r \equiv \langle\!\langle \mathbf{v} \rangle\!\rangle (\neg \mathsf{neg}\,\mathcal{U}(\mathsf{sat} \vee (\mathsf{neg} \wedge \langle\!\langle \emptyset \rangle\!\rangle \bigcirc \neg\varphi_{r-1}))$$

for $r = 1, \ldots, p$.

Before we come to the theorem proving the reduction we state a fundamental lemma which can be seen as a counterpart of Lemma 10.47.

**Lemma 10.52.** *Let $I = (\varphi_1, \ldots, \varphi_p)$ be an $\text{SNSAT}_1$ instance.*

*(a) Let $T = (T_1, \ldots, T_p)$ be a solution for $I$. For all $r = 1, \ldots, p$, if $z_r \in T_r$ then there is a $\mathbf{v}$-choice of $\mathfrak{M}(\varphi_r)$ such that for the set $L$ of all literal states reachable from $q_0^r$ and which belong to $\mathfrak{M}(\varphi_r)$ it holds that $\{x \in X^r \cup Z \mid q_w \in L \text{ and } \chi_w^{\varphi_r} = x\} \subseteq T_r$ and $\{x \in X^r \cup Z \mid q_w \in L \text{ and } \chi_w^{\varphi_r} = \neg x\} \cap T_r = \emptyset$.*

*(b) Let $I^{p-1} = (\varphi_1, \ldots, \varphi_{p-1})$ and let $T^{p-1} = (T_1^{p-1}, \ldots, T_{p-1}^{p-1})$ be an $I^{p-1}$-witness. Then, if there is a $\mathbf{v}$-choice of $\mathfrak{M}(\varphi_p)$ such that for the set $L$ of all literal states reachable from $q_0^p$ that belong to $\mathfrak{M}(\varphi_p)$ we have that*

*(i) for any $q_w, q_v \in L$ the literals $\chi_w^{\varphi_p}$ and $\chi_v^{\varphi_p}$ are non-complementary; and*

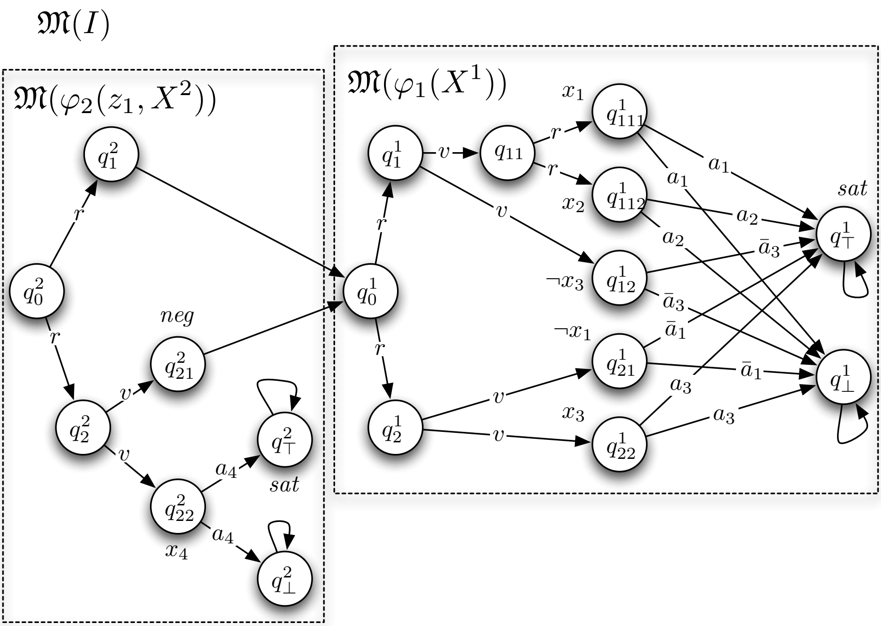



**Fig. 10.8.** The construction of $\mathfrak{M}(I)$ for $\varphi_1 = ((x_1 \wedge x_2) \vee \neg x_3) \wedge (\neg x_1 \vee x_3)$ and $\varphi_2 = z_1 \wedge (\neg z_1 \vee x_4)$. Transitions labeled with an agent indicate that this agent can force the transition disregarding the behaviour of the other agents.

*(ii) if $q_v \in L$ with $\chi_v^{\varphi_p} = z_i$ (resp. $\chi_v^{\varphi_p} = \neg z_i$) then $z_i \in T_i$ (resp. $z_i \notin T_i$); then, $T = (T_1^{p-1}, \ldots, T_{p-1}^{p-1}, T_p)$ is a solution for $I$ where*

$$T_p = \{x \in X^r \mid q_w \in L \textit{ and } \chi_w = x\} \cup \{z^i \mid z^i \in T_i^{p-1},\ i < p\} \cup \{z^p\}.$$

*(c) $\mathfrak{M}(I), q_0^i \models \varphi_i$ if, and only if, $\mathfrak{M}(I), q_0^i \models \varphi_j$; and $\mathfrak{M}(I), q_0^i \models \neg\varphi_i$ if, and only if, $\mathfrak{M}(I), q_0^i \models \neg\varphi_j$ for all $j \geq i$.*

The complete proof is given on page 319.

Similar to [Bulling et al., 2009b; Jamroga and Dix, 2008], we have the following result which shows that the construction above is a polynomial-time reduction of $\mathrm{SNSAT}_1$ to model checking **CoalATL**.

**Theorem 10.53.** *The size of $\mathfrak{M}(I)$ and of the formulae $\varphi_p$ is polynomially in the size of the $\mathrm{SNSAT}_1$ instance $I = (\varphi_1, \ldots, \varphi_p)$ and we have the following:*

*There is a solution $T = (T_1, \ldots, T_r)$ of $I^r = (\varphi_1, \ldots, \varphi_r)$ if, and only if,*
$$\mathfrak{M}(I^r), q_0^r \models \varphi_l$$

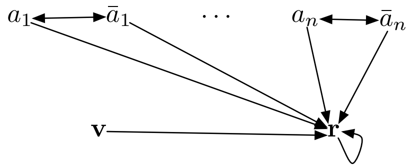



**Fig. 10.9.** Proof of Corollary 10.56.

*for $l \geq r$ and $r \leq p$.*

The complete proof is given on page 320.

The reduction gives us the following hardness result.

**Theorem 10.54.** *Model checking* **CoalATL** *is* $\mathbf{\Delta_2^P}$*-hard for any reduction-suitable semantics.*

With Corollary 10.38 we obtain the following completeness result.

**Theorem 10.55.** *Model checking* **CoalATL** *is* $\mathbf{\Delta_2^P}$*-complete for any reduction-suitable semantics* $\mathfrak{sem}$ *with* $\mathcal{VER}_{\mathfrak{sem}} \in \mathbf{NP}$.

Finally, we show completeness for the stable semantics.

**Corollary 10.56.** *Model checking* **CoalATL** *is* $\mathbf{\Delta_2^P}$*-complete for the* $\mathfrak{sem}_{stable}$ *semantics.*

*Proof.* We have to show that the semantics is reduction-suitable; that is, we have to construct a $\mathfrak{sem}$-witness over $\mathbb{A}\mathrm{gt}$.

Consider the coalitional framework shown in Figure 10.9. We show that this is a witness for the reduction suitableness of the stable semantics. Let $X = \{x_1, \ldots x_n\}$ and $\bar{X} = \{\bar{x}_1, \ldots, \bar{x}_n\}$ be given and $f(x_i) := \bar{x}_i$. We show that the conditions from Definition 10.43 are met.

"$\Rightarrow$": Let $E \in \mathfrak{sem}_{\text{stable}}$. Clearly, $\mathbf{r} \notin E$ because $\{\mathbf{r}\}$ is not conflict-free. Moreover, $\{\mathbf{v}\} \in E$; as $\{\mathbf{v}\}$ cannot be attacked by $E$. We also have that $E \subseteq X \cup \bar{X}$. Suppose there is an $i$ such that $\{a_i, \bar{a}_i\} \cap E = \emptyset$. Then, $\{a_i, \bar{a}_i\}$ is not attacked by $E$. Contradiction. Now suppose that $\{a_i, \bar{a}_i\} \subseteq E$. This is a contradiction since $\{a_i, \bar{a}_i\}$ is not conflict-free.

"$\Leftarrow$": Suppose $E \cup \{\mathbf{v}\}$ satisfies condition (1), (2), and (3). Clearly, $E \cup \{\mathbf{v}\}$ is conflict-free. Because for each $i$ either $a_i$ or $\bar{a}_i$ is in $E \cup \{\mathbf{v}\}$ every element outside is attacked by some element from $E \cup \{\mathbf{v}\}$.

■

| Types of Semantics | MC Complexity |
|---|---|
| $\mathcal{VER}_{\mathfrak{sem}} \in \mathcal{C}$ | $\mathbf{P}^{\mathbf{NP}^{\mathcal{C}}}$ |
| $\mathfrak{sem}_{\text{preferred}}$ | $\mathbf{\Delta_3^P}$ |
| reduction-suitable | $\mathbf{\Delta_2^P}$-hard |
| $\mathcal{VER}_{\mathfrak{sem}} \in \mathbf{NP}$, reduction-suitable | $\mathbf{\Delta_2^P}$-complete |
| $\mathfrak{sem}_{\text{stable}}$ | $\mathbf{\Delta_2^P}$-complete |
| $\mathcal{VER}_{\mathfrak{sem}} \in \mathbf{NP}$ | $\mathbf{\Delta_2^P}$ |
| $\mathfrak{sem}_{\text{admissible}}$, $\mathfrak{sem}_{\text{complete}}$ | $\mathbf{\Delta_2^P}$ |
| polynomially many coalitions, polynomially enumerable $\mathfrak{sem}$ | $\mathbf{P}$-complete |
| $\mathfrak{sem}_{\text{grounded}}$ | $\mathbf{P}$-complete |

**Fig. 10.10.** Overview of the model checking results modulo the complexity of the used argumentation semantics.

## 10.3 CSLP

In this section we discuss the model checking complexity of **CSLP** and $\mathbf{CSLP}^1$. The model checking algorithm is derived from combining the results for **CSL** and **ATLP**. We will just consider the case for $\mathbf{CSLP}^{\mathbf{base}}$ and $\mathbf{CSLP}^1$; all the other cases that do not rely one the classical polynomial time algorithm of **ATL** model checking are obtained similarly to the results for **ATLP** from Section 10.1.

As before, in the following we use $l$ to refer to the length of $\varphi$ and $m$ to denote the number of transitions in $\mathfrak{M}$. We only consider a restricted class of models in which the check for plausibility of a strategy profile can be done in polynomial time (wrt $l$ and $m$) by a non-deterministic Turing machine. In order to conduct a sensible analysis such an assumption is necessary. To this end, we adapt the important notion of well-behaved CGSP from Definition 10.2

**Definition 10.57 (Well-behaved** ICGSP**).** *A* ICGSP $\mathfrak{M}$ *is called* well-behaved *if, and only if, (1)* $\Upsilon_{\mathfrak{M}} = \Sigma$*: all the strategy profiles are plausible in* $\mathfrak{M}$*; and (2) there is a non-deterministic Turing machine which determines whether* $s \in [\![\omega]\!]_{\mathfrak{M}}^{Q'}$ *for every set* $Q' \subseteq Q_{\mathfrak{M}}$*, strategy profile* $s \in \Sigma$*, and plausibility term* $\omega \in \Omega$ *in polynomial time wrt the length of* $\omega$ *and the number of transitions in* $\mathfrak{M}$*.*

The results for **ATLP** are given in Section 10.1. We begin by reviewing the existing results for **CSL**. The complexity results for **CSLP** follow in a natural way. In [Jamroga and Ågotnes, 2007] it was shown that **CSL** model checking is $\mathbf{\Delta_2^P}$-complete, the hard cases being formulae $\langle\!\langle A\rangle\!\rangle\Box\varphi$ and $\langle\!\langle A\rangle\!\rangle\varphi_1\mathcal{U}\varphi_2$. The formulae require the existence of a single uniform strategy which is successful in *all* states of $Q'$. In the algorithm from [Jamroga and Ågotnes, 2007], the strategy is guessed by the oracle and then verified in polynomial time. Nested cooperation modalities are model-checked recursively (bottom-up) which puts the algorithm indeed in $\mathbf{\Delta_2^P}$.

We also recall from Section 10.1 that $\mathsf{ATLP}^{\mathsf{base}}$ model checking is $\mathbf{\Delta_3^P} = \mathbf{P}^{\mathbf{NP}^{\mathbf{NP}}}$-complete. The algorithm for checking the hard cases ($\langle\langle A\rangle\rangle\Box\varphi$ and $\langle\langle A\rangle\rangle\varphi_1\mathcal{U}\varphi_2$) is similar: Firstly, a *plausible* strategy of $A$ is guessed (first **NP**-oracle call) and verified against all *plausible* strategies of the opponents (second **NP**-oracle call, the "worst" response of the opponents is guessed). Note that, as soon as the relevant strategy (or strategy profile) $s$ is fixed, the remaining verification can be done in deterministic polynomial time: it is enough to "trim" the model by deleting all transitions which cannot occur when the agents follow $s$, and to model check a $\mathsf{CTL}$ formula in the trimmed model (which can be done in polynomial time [Emerson, 1990][Clarke et al., 1986]).

For $\mathsf{CSLP}^{\mathsf{base}}$, we essentially use the $\mathsf{ATLP}^{\mathsf{base}}$ model checking algorithm with an additional check for uniformity of strategies. This does not influence the complexity. We obtain the following result (we refer to Section 10.1 and to [Jamroga and Ågotnes, 2007] for details).

**Theorem 10.58.** *Model checking* $\mathsf{CSLP}^{\mathsf{base}}$ *in the class of well-behaved* ICGSP *is* $\mathbf{\Delta_3^P}$*-complete with respect to* $l$ *and* $m$.

*Proof.* [Sketch] The hardness follows from the fact that $\mathsf{ATLP}^{\mathsf{base}}$ is $\mathbf{\Delta_3^P}$-complete and can be embedded in $\mathsf{CSLP}$ (cf. Proposition 7.28). For the inclusion in $\mathbf{\Delta_3^P}$, we sketch the algorithm for $\mathfrak{M}, Q' \models \langle\langle A\rangle\rangle\Box\varphi$: (1) Model-check $\varphi$ recursively for each $q \in Q_{\mathfrak{M}}$, and label the states for which $\mathfrak{M}, q \models \varphi$ with a new proposition $\mathsf{p}$; (2) Guess a "good" plausible uniform strategy $s_A$; (3) Guess a "bad" uniform plausible strategy profile $t$ such that $t|_A = s_A$; and (4) Return true if $Q' \subseteq mcheck_{\mathsf{CTL}}(\mathfrak{M}', \mathsf{A} \bigcirc \mathsf{p})$ and false otherwise, where $\mathfrak{M}'$ is the trimmed model of $\mathfrak{M}$ wrt profile $t$. ■

In Section 7.3 we showed how $\mathsf{CSLP}^1$ can be used to characterise incomplete information solution concepts. However, for this reason we had to use the extended language of $\mathsf{CSLP}^1$. An obvious question arises: How much does the complexity increase? The answer is quite appealing: The increase in complexity depends on how much extra-expressiveness we actually use; and in any case, we get some expressiveness for free! This can be shown analogously to $\mathsf{ATLP}^1$. Similarly, the model checking complexity can be completely characterised in the number of quantifier *alternations* used in the extended plausibility terms. If we have no quantifiers at all, the resulting sublanguage is no more costly to verify than the base version. Note that the quantifier-free sublanguage of $\mathsf{CSLP}^1$ is already sufficient to "plug in" important solution concepts (e.g., Nash equilibria). For each additional quantifier alternation (starting with a universal quantifier) the complexity is pushed one level up in the polynomial hierarchy. The following result is shown analogously to Theorem 10.30. We use the same notation as in Section 10.1; in particular, the notion of level-$i$ formulae from Definition 10.27

**Theorem 10.59.** *Let $\varphi$ be an level-i formula of $\mathcal{L}^1_{CSLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$, $\mathfrak{M}$ be a well-behaved* ICGS, *and $Q'$ a set of states. Then, model checking $\mathfrak{M}, Q' \models \varphi$ can be done in time $\mathbf{\Delta^P_{i+5}}$ with respect to $l$ and $m$.*

*Remark 10.60.*

(a) We would like to remark once more that all the other results from Section 10.1 that do not rely one the classical polynomial time algorithm of **ATL** model checking do also hold mutatis mutantis. But, the polynomial time results over pure CGSs and rectangular CGSPs, given in Section 10.1, do for example *not* hold for **CSLP**.
(b) In particular, we would like to note that the model checking algorithm for $\mathcal{L}^1_{CSLP}(\mathbb{A}\mathrm{gt}, \Pi, \emptyset)$ over pure ICGS (i.e without hard-wired plausibility terms) can also only be shown to be in $\mathbf{\Delta^P_{i+5}}$. That is, Theorem 10.29 cannot be applied.
(c) Due to the flaw in the calculation of the nesting level pointed out in Remark 10.26 also the model checking result from [Bulling and Jamroga, 2009b] are affected. We can only prove the upper bound $\mathbf{\Delta^P_{i+5}}$ instead of $\mathbf{\Delta^P_{i+3}}$.

## 10.4 **ATL** with Probabilistic Success

In this section, we discuss the complexity of model checking the logics **pATL**$_{MMS}$ and **pATL**$_{BS}$.

We have presented two alternative semantics for the logic, underpinned by two different ways of assuming the opponents' behaviour. The semantics based on mixed strategies seems to be the simpler of the two, as the success measure is based on a finite probability distribution, and hence can be computed as a finite sum of elements. In contrast, the semantics based on behavioural strategies refers to an integral of a continuous probability distribution – so one might expect that checking formulae $\mathcal{L}_{pATL}$ in the latter case is much harder. Surprisingly, it turns out that the reality is completely opposite.

### 10.4.1 **pATL**$_{MMS}$

We study the model checking problem with respect to the number of transitions in the model ($m$) and the length of the formula ($l$). As the number of memoryless strategies is usually exponential in the number of transitions, we need a compact way of representing mixed strategies (representing them

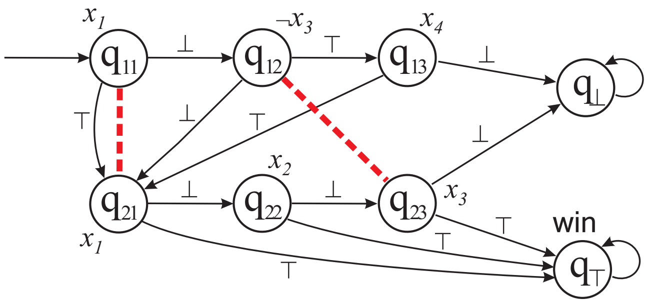



**Fig. 10.11.** The concurrent epistemic game structure for formula $F \equiv (x_1 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee x_3)$. States $q_{11}, q_{21}$ and $q_{12}, q_{23}$ are indistinguishable for the agent: the same action (valuation) must be specified in both within a uniform strategy.

explicitly as arrays of probability values would yield structures of exponential size). For the rest of this section, we assume that a mixed strategy is represented as a sequence of pairs $[\langle \mathcal{C}_1, p_1\rangle, \dots, \langle \mathcal{C}_n, p_n\rangle]$, where the length of the sequence is polynomial in $m$, and $l$, every $\mathcal{C}_i$ is a condition on strategies that can be checked in polynomial time wrt $m, l$, and every $p_i \in [0, 1]$ is a probability value with a polynomial representation wrt $m, l$. For simplicity, we assume that conditions $\mathcal{C}_i$ are mutually exclusive. The idea is that the probability of strategy $s$ is determined as $p(s) = p_i$ by the condition $\mathcal{C}_i$ which holds for $s$; if no $\mathcal{C}_i$ holds for $s$ then the probability of $s$ is 0. We also assume that the distribution is normalised, i.e., $\sum_{s\in\Sigma} p(s) = 1$ where $p(s)$ denotes the probability of $s$ determined by the representation given above.

In this setting, model checking $\mathcal{L}_{pATL}$ with mixed memoryless strategies turns out to be at least **PP**-hard, where **PP** ("Probabilistic Polynomial time") is the class of decision problems solvable by a probabilistic Turing machine in polynomial time, with an error probability of less than 1/2 for all instances [Gill, 1977] (cf. Definition 4.4). We prove it by a polynomial-time reduction of "Majority SAT" (see Definition 4.17), a typical **PP**-complete problem. Since **PP** contains both **NP** and co-**NP** [Beigel et al., 1995] (cf. Theorem 4.6), we obtain **NP**-hardness and co-**NP**-hardness as an immediate corollary.

**Proposition 10.61.** *Model checking* $\mathbf{pATL}_{MMS}$ *is* **PP***-hard.*

*Proof.* We prove hardness by a reduction of MAJSAT. First, we take the Boolean formula $F$ and construct a single agent concurrent epistemic game structure $\mathfrak{M}$ in a way similar to [Schobbens, 2004]. The model includes 2 special states: $q_\top$ (the winning state) and $q_\perp$ (the losing state), plus one state for each literal instance in $F$. The "literal" states are organised in levels, according to the clause they appear in: $q_{ij}$ refers to the $j$th literal of clause $i$.

At each “literal” state, the agent can declare the underlying proposition true or false. If the declaration validates the literal, then the system proceeds to the next clause; otherwise it proceeds to the next literal in the same clause. For example, if $q_{12}$ refers to literal $\neg x_3$, then action “true” makes the system proceed to $q_{13}$ (in search of another literal that would validate clause 1), while action “false” changes the state to $q_{21}$ (to validate the next clause). In case the last literal in a clause has been invalidated, the system proceeds to $q_\bot$; when a literal in the last clause is validated, a transition to $q_\top$ follows. There is a single atomic proposition $\mathsf{win}$ in the model, which holds only in state $q_\top$. An example of the construction is shown in Figure 10.11.

Two nodes with the same underlying proposition are connected by an indistinguishability link to ensure that strategies consistently assign variables $x_1, \ldots, x_n$ with Boolean values. To achieve this, it is enough to require that only *uniform* strategies are used by the agent; a strategy is uniform iff it specifies the same choices in indistinguishable states. Now we observe the following facts:

- There is a 1-to-1 correspondence between assignments of $x_1, \ldots, x_n$ and uniform strategies of the validating agent. Also, each uniform strategy $s$ determines exactly one path $\lambda(s)$ starting from $q_{11}$;
- By the above, the number of uniform strategies is equal to the number of different assignments of $x_1, \ldots, x_n$. Thus, there are $D = 2^n$ uniform strategies in total;
- A uniform strategy successfully validates $F$ iff it enforces path $\lambda(s)$ that achieves $q_\top$, i.e., one for which $\lambda(s) \models \Diamond\mathsf{win}$;
- Uniformity of a strategy can be checked in time polynomial wrt $m$ (the number of transitions in the model). Let $\mathcal{C}$ be an encoding of the uniformity condition; then, mixed strategy $[\langle \mathcal{C}, \frac{1}{D} \rangle]$ assigns the same importance to every uniform strategy and discards all non-uniform ones. We define symbol $\omega$ to denote that strategy.

Finally, we have the following reduction:

MAJSAT(F)=YES

iff $\frac{\#\text{ assignments } V \text{ of } x_1, \ldots, x_n \text{ such that } V \models F}{\#\text{ all assignments of } x_1, \ldots, x_n} > 0.5$

iff $\frac{\#\text{ uniform strategies } s \text{ such that } \lambda(s) \models \Diamond \mathsf{win}}{\#\text{ all uniform strategies}} > 0.5$

iff $\frac{\#\text{ uniform strategies } s \text{ such that } \lambda(s) \not\models \Diamond \mathsf{win}}{\#\text{ all uniform strategies}} < 0.5$

iff not $\frac{\#\text{ uniform strategies } s \text{ such that } \lambda(s) \not\models \Diamond \mathsf{win}}{\#\text{ all uniform strategies}} \geq 0.5$

iff not $\frac{\#\text{ uniform strategies } s \text{ such that } \lambda(s) \models \Box \neg \mathsf{win}}{\#\text{ all uniform strategies}} \geq 0.5$

iff not $success(s_\emptyset, [\![\omega]\!]_1, q_{11}, \Box\neg\mathsf{win}) \geq 0.5$

iff $\mathfrak{M}, q_{11} \models \neg\langle\langle\emptyset\rangle\rangle^{0.5}_{\omega}\Box\neg\mathsf{win}$.

■

**Corollary 10.62.** *Model checking* **pATL**$_{MMS}$ *is* **NP***-hard and co-***NP***-hard.*

For the upper bound, we present a **PSPACE** algorithm for model checking **pATL**$_{MMSs}$. The algorithm uses an $\mathbf{NP}^{\#\mathbf{P}}$ procedure, i.e., one which runs in nondeterministic polynomial time with calls to an oracle that counts the number of accepting paths of a nondeterministic polynomial time Turing machine [Valiant, 1979]. The class $\mathbf{NP}^{\#\mathbf{P}}$ is known to lie between **PH** and **PSPACE** [Toda, 1989; Angluin, 1980]. That $\mathbf{PH} \subseteq \mathbf{NP}^{\#\mathbf{P}}$ follows from Theorem 4.7 and Theorem 4.9. That $\mathbf{NP}^{\#\mathbf{P}} \subseteq \mathbf{PSPACE}$ follows from $\mathbf{P}^{\#\mathbf{P}} \subseteq \mathbf{PSPACE}$ and the observation that $\mathbf{NP}^{\#\mathbf{P}} \subseteq \mathbf{NP}^{\mathbf{P}^{\#\mathbf{P}}} \subseteq \mathbf{NP}^{\mathbf{PSPACE}} = \mathbf{PSPACE}$.

**Theorem 10.63.** *Model checking* **pATL**$_{MMS}$ *is in* **PSPACE** *for the class of* pCGSs *which allows a presentation of the predicted behaviour as introduced above.*

*Proof.* Let $\gamma$ be a path formula that does not include cooperation modalities. The following procedure checks if $\mathfrak{M}, q \models \langle\langle A\rangle\rangle^{p}_{\sigma}\gamma$:

1. Nondeterministically choose a strategy $s_A$ of agents $A$; /requires at most $m$ steps/
2. For each $\langle \mathcal{C}_i, p_i\rangle \in [\![\sigma]\!]$, execute $T_i := oracle(s_A, \mathcal{C}_i)$; /polynomially many calls/
3. Answer YES if $\sum_i p_i T_i \geq p$ and NO otherwise. /computation polynomial in the representation of $p_i$ and $T_i$/

The oracle computes the number of $\mathbb{A}\mathrm{gt} \setminus A$'s strategies $t_{\mathbb{A}\mathrm{gt}\setminus A}$ such that $t_{\mathbb{A}\mathrm{gt}\setminus A}$ obeys $\mathcal{C}_i$ and $\langle s_A, t_{\mathbb{A}\mathrm{gt}\setminus A}\rangle$ generate a path that satisfies $\gamma$. That is, the oracle counts the accepting paths of the following nondeterministic Turing machine:

1. Nondet. choose a strategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$ of agents $\mathbb{A}\mathrm{gt} \setminus A$; /requires at most $m$ steps/
2. Check whether $t_{\mathbb{A}\mathrm{gt}\setminus A}$ satisfies $\mathcal{C}_i$; /polynomially many steps/
3. If so, "trim" model $\mathfrak{M}$ by removing choices that are not in $\langle s_A, t_{\mathbb{A}\mathrm{gt}\setminus A}\rangle$, then model-check the **CTL** formula $\mathsf{A}\gamma$ in the resulting model and return the answer of that algorithm; otherwise return NO. /$m$ steps + CTL model checking which is polynomial in $m, l$ [Clarke and Emerson, 1981]/

The main procedure runs in time $\mathbf{NP}^{\#\mathbf{P}}$, and hence the task can be done in polynomial space. For the case when $\gamma$ includes nested strategic modalities, the procedure is applied recursively (bottom-up). That is, we get a deterministic Turing machine with adaptive calls to the **PSPACE** procedure. Since $\mathbf{P}^{\mathbf{PSPACE}} = \mathbf{PSPACE}$, we obtain the upper bound. ■

### 10.4.2 Model Checking pATL$_{BS}$

The semantics of **pATL**$_{BS}$ with opponents' behaviour modelled by behavioural strategies is mathematically more advanced than for mixed strategies. So, one may expect the corresponding model checking problem to be even harder than the one we studied in Section 10.4.1. Surprisingly, it turns out that checking **pATL**$_{BS}$ can be done in polynomial time wrt the number of transitions in the model ($m$) and the length of the formula ($l$). Below, we sketch the procedure $mcheck(M, q, \varphi)$ that checks whether $\mathfrak{M}, q \models \varphi$:

- $\varphi \equiv p$, $\neg\psi$, or $\psi_1 \wedge \psi_2$: proceed as usual;
- $\varphi \equiv \langle\langle A\rangle\rangle^{p}_{\sigma}\Box\psi$: (for $\varphi \equiv \langle\langle A\rangle\rangle^{p}_{\sigma} \bigcirc \psi$ and $\varphi \equiv \langle\langle A\rangle\rangle^{p}_{\sigma}\psi_1\mathcal{U}\psi_2$ analogously)
  1. Model check $\psi$ in $\mathfrak{M}$ recursively. Replace $\psi$ with a new proposition $\mathsf{yes}$ holding in exactly those states $st \in Q$ for which $mcheck(M, st, \psi) =$ YES;
  2. Reconstruct $\mathfrak{M}$ as a 2-player CGSP $\mathfrak{M}'$ with agent 1 representing team $A$ and 2 representing $\mathbb{A}\mathrm{gt} \setminus A$. That is, $d'_1(st) = \prod_{a\in A} d_a(st)$, $d'_2(st) = \prod_{a\in \mathbb{A}\mathrm{gt}\setminus A} d_a(st)$ for each $st \in Q$, and the transition function $o'$ is updated accordingly.
  3. Fix the behaviour of agent 2 in $\mathfrak{M}'$ according to $[\![\omega]\!]_{\mathbb{A}\mathrm{gt}\setminus A}$. That is, construct the probabilistic transition function $o''$ so that, for each $st, st' \in Q, \alpha_1 \in d'_1(st)$: $o''(st, \alpha_1, st') = \sum_{\{\alpha_2\in d'_2(st) \mid o'(st,\alpha_1,\alpha_2)=st'\}} [\![\omega]\!]_{\mathbb{A}\mathrm{gt}\setminus A}(st, \alpha_2)$. Also, reconstruct proposition $\mathsf{yes}$ as a reward function that assigns 1 at state $st$ if $\mathsf{yes} \in \pi'(st)$ and 0 otherwise. Note that the resulting structure $\mathfrak{M}''$ is a Markov Decision Process [Bellman, 1957];
  4. Model check the formula $\exists\Box\mathsf{yes}$ of "Discounted **CTL**" [de Alfaro et al., 2004] in $\mathfrak{M}'', q$ and return the answer. This can be done in time polynomial in the number of transitions in $\mathfrak{M}''$ and exponential in the length of the formula [de Alfaro et al., 2004]. Note, however, that the length of $\exists\Box\mathsf{yes}$ is constant.

As parts 2-4 require $O(m)$ steps, and they are repeated at most $l$ times (once per subformula of $\varphi$), we get that the procedure runs in time $O(ml)$.

For the lower bound, we observe that reachability in And-Or-Graphs [Immerman, 1981] can be reduced (in constant time) to model checking of the fixed $\mathcal{L}_{ATL}$ formula $\langle\langle a\rangle\rangle\Diamond p$ over acyclic CGSs (cf. [Alur et al., 2002]). Alternatively, one can reduce the Circuit Value Problem [Vollmer, 1999] to **ATL** model checking over acyclic CGSs in a similar way. By Proposition 7.58, this reduces (again in constant time) to model checking of $\mathbf{pATL}_{BS}$. Therefore, we get the following result.

**Theorem 10.64.** *Model checking* $\mathbf{pATL}_{BS}$ *is* **P***-complete with respect to the number of transitions in the model and the length of the formula.*

Thus, it turns out that the model checking problem associated with the more sophisticated semantics can be done in linear time wrt the input size, while model checking the seemingly simpler semantics is much harder (**NP**- and co-**NP**-hard).

# 11

# Verifying Resource-Bounded Agents

## Contents



In this chapter we consider the complexity of the model checking problem of the resource-bounded logics introduced in Chapter 8. We show that the single-agent case is decidable for the least expressive version of the resource-bounded tree logics and that the multi-agent settings are undecidable in general.

## 11.1 Model Checking $\mathsf{RTL}^\star$ in Restricted Settings

We are mainly interested in the verification of systems. *Model checking* refers to the problem whether a formula $\varphi$ is true in an RBM $\mathfrak{M}$ and a state $q$ in $\mathfrak{M}$. For $\mathsf{CTL}^\star$ this problem is **PSPACE**-complete and for $\mathsf{CTL}$, the fragment of $\mathsf{CTL}^\star$ in which each temporal operator is directly preceded by a path quantifier, it is **P**-complete [Clarke et al., 1986]. So, we cannot hope for our problem to be computationally any better than **PSPACE** in the general setting; actually, it is still open whether it is decidable at all.

In the following, we consider the decidability of fragments of the full logic over special classes of RBMs (which of course, implies decidability of the restricted version over the same class of models).

**Proposition 11.1 (Decidability: Production -, zero free).** *The model checking problem for* $\mathsf{RTL}^\star{}_{\mathcal{R}es^\pm}$ *over production- and zero-free* RBMs *is decidable.*

*Proof.* Let $\langle\rho\rangle\gamma$ be given where $\gamma$ does not contain any more path quantifiers. According to Prop. 8.23(a) all $\rho$-feasible paths have the form $\lambda' \circ (q)^\omega$. Lemma 8.33 allows us to restrict to a finite set of such paths for a given $\gamma$: For any path $\lambda$ we consider $\lambda^{[|\gamma|]}$. Therefore, there are only finitely many $\rho$-feasible paths of interest for $\rho \in \mathcal{R}es^\pm$ and $\gamma$.

This set can be computed step by step. In order to verify whether $\mathfrak{M}, q \models \langle\rho\rangle\gamma$ it is necessary to check whether $\gamma$ holds on any of the finitely many $\rho$-feasible relevant paths starting in $q$. The model checking algorithm proceeds bottom-up. ■

From Corollary 8.32 we know that we can use a $\mathsf{CTL}^\star$ model checker over $k$-bounded models.

**Proposition 11.2 (Decidability: $k$-bounded).** *The model checking problem for* $\mathsf{RTL}^\star{}_{\mathcal{R}es^\pm}$ *over $k$-bounded* RBMs *is decidable and* **PSPACE***-hard.*

By Prop. 8.25 and the observation that resources with an infinite quantity can be neglected in a production-free RBM we have the following theorem.

**Theorem 11.3 (Decidability: production free).** *The model checking problem for* $\mathsf{RTL}^\star$ *over production-free* RBMs *is decidable and* **PSPACE***-hard.*

## 11.2 Model Checking RTL is Decidable

The following result shows that model checking **RTL** is decidable.

### 11.2.1 RBMs and Petri Nets

The main idea is to encode an RBM as a Petri net and then to use decision procedures for Petri nets to solve the model checking problem.

We can encode an RBM $\mathfrak{M}$ with respect to a given set $Q' \subseteq Q_{\mathfrak{M}}$, and a feasible resource set $\rho$ as a Petri net $N_{Q',\rho}(\mathfrak{M}) = (S, T, W, m^I)$. The main idea of encoding transitions is sketched in Figure 11.1. States $q$ are encoded

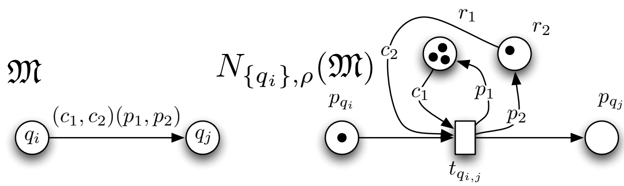



**Fig. 11.1.** Petri-net encoding $N_{\{q_i\},\rho}(\mathfrak{M})$ of an RBM $\mathfrak{M}$. Tokens inside the places $r_k$ represent the amount of that resource (i.e., $\rho(r_1) = 3$ and $\rho(r_2) = 1$). Outgoing paths consume tokens and incoming paths produce tokens, labeled edges produce/consume the amount the edge is annotated with. E.g., if there is a token in place $p_{q_i}$ and $c_k$ tokens in place $r_k$ then the token can be moved to $p_{q_j}$ and $p_k$ tokens can be moved to $r_k$ for $k = 1, 2$.

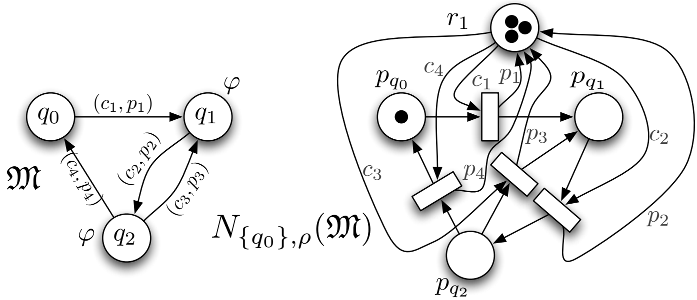



**Fig. 11.2.** Example of a complete encoding of an RBM $\mathfrak{M}$ where $\rho(r_1) = 3$.

as places $p_q$ and transitions between states as transitions between places. For each resource type a new place is created. For the initial marking function $m^I$ we have that $m^I(p_q) = 1$ for all $q \in Q'$, $m^I(r) = \rho(r)$ for $r \in \mathcal{R}es$, and 0 otherwise. A complete encoding of an RBM is shown in Figure 11.2. We denote (the unique) transition between place $p_{q_i}$ and $p_{q_j}$ by $t_{q_i q_j}$. (We are economical with our notation and reuse $t$ already known from RBMs.)

**Lemma 11.4.** *Let $\rho$ be a feasible resource set, $\mathfrak{M}$ an* RBM*, and $q \in Q_{\mathfrak{M}}$. Then, the following holds:*

*$q_0 q_1 \ldots$ is a $\rho$-feasible path in $(\mathfrak{M}, q)$ iff $\sigma = t_{q_0 q_1} t_{q_1 q_2} \ldots$ is a run in $N_{\{q_0\},\rho}(\mathfrak{M})$.*

*Proof.* The proof is done by induction on the length $i$ of the path and run, respectively. Consider the case $i = 2$. Suppose the transition $(q_0, q_1)$ is $\rho$-feasible and $\eta_1$ resources are available in $q_1$. By construction each resource state $r_i$ contains $\rho(r_i)$ tokens and there is a token in $p_{q_0}$; thus, transition $t_{q_0 q_1}$ can fire. Clearly, the tokens in the resource places (i.e. $m_1$) match with the

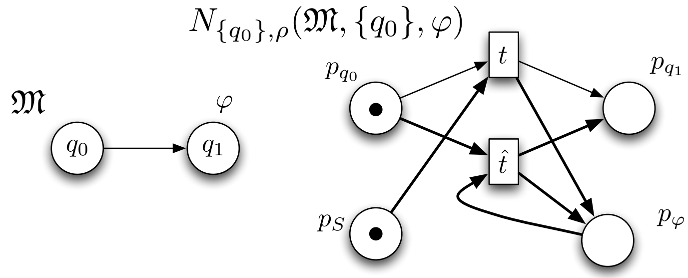



**Fig. 11.3.** The encoding $N_{\{q_0\},\rho}(\mathfrak{M}, \{q_0\}, \varphi)$ of an RBM $\mathfrak{M}$. The resource requirements are left out here.

resources available in $q_1$. Now suppose the claim is correct up to position $j$. Then, it is easily seen that a transition $q_j \to q_{j+1}$ is feasible iff the transition $t_{q_j q_{j+1}}$ can fire. ■

In order to model check specific formulae, we need to extend our encoding. For example, consider the formula $\langle\rho\rangle\Diamond\varphi$ where $\varphi$ is a propositional formula and $\rho$ a feasible resource set. We can decompose the model checking problem into two parts:

1. Find a (finite) sequence of states feasible given $\rho$ to a state in which $\varphi$ holds; and
2. then arbitrarily extend this (finite) sequence to an infinite $\rho$-feasible path.

To achieve this, we introduce a new place that indicates (by marking it with a token) that $\varphi$ has been made true. This place remains marked throughout the subsequent executions of the net and hence serves as an indicator of item 1 having been satisfied. To achieve this, given a propositional formula $\varphi$ we extend the encoding $N_{\{q_0\},\rho}(\mathfrak{M})$ of $\mathfrak{M}$ to an encoding $N_{\{q_0\},\rho}(\mathfrak{M}, Q', \varphi)$ where $Q' \subseteq Q$ as explained in the following. The new Petri net is equal to $N_{\{q_0\},\rho}(\mathfrak{M})$ apart from the following modifications (Figure 11.3 illustrates the construction):

1. $N'$ has two new places $p_S$ and $p_\varphi$.
2. For each transition $t$ in $N(\mathfrak{M})$ that corresponds to a transition $q \to q'$ in $\mathfrak{M}$ such that $q \in Q'$ and $q' \models^{\text{prop}} \varphi$ we construct a duplicate with the fresh name $\hat{t}$ and include the following arcs: $p_S$ is connected to $t$; $t$ and $\hat{t}$ are connected to $p_\varphi$; and $p_\varphi$ is also connected to $\hat{t}$; i.e. $W(p_S, t) = W(t, p_\varphi) = W(p_\varphi, \hat{t}) = W(\hat{t}, p_\varphi) = 1$.
3. $p_S$ is initially marked.

The following proposition is clear from the construction of the net $N_{\{q_0\},\rho}(\mathfrak{M}, \{q_0\}, \varphi)$.

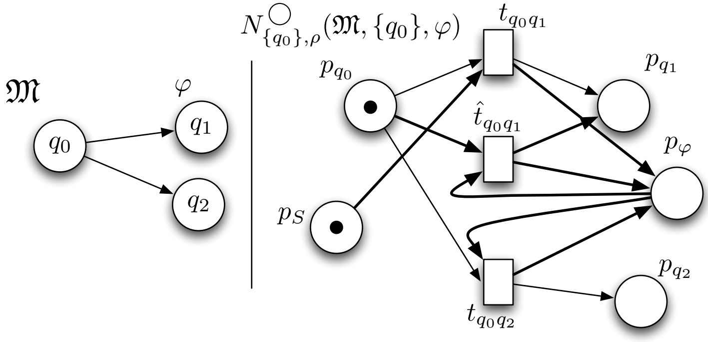



**Fig. 11.4.** The encoding $N^{\bigcirc}_{\{q_0\},\rho}(\mathfrak{M}, \{q_0\}, \varphi)$ of an RBM $\mathfrak{M}$. The resource requirements are left out here.

**Proposition 11.5.** *The constructed Petri net $N_{\{q_0\},\rho}(\mathfrak{M}, \{q_0\}, \varphi)$ has the following properties:*

1. *A transition $t$ can only be enabled if there is a token in $p_S$.*
2. *Once such a transition $t$ has fired it can never be enabled again and there is a token in $p_\varphi$*
3. *A transitions $\hat{t}$ can only be enabled if there is a token in $p_\varphi$.*
4. *Once there is a token in $p_\varphi$ it remains there forever.*
5. *$p_S$ and $p_\varphi$ contain at most one token and there is a token in $p_S$ iff there is no token in $p_\varphi$.*

Additionally, for the next-operator we extend the construction and disable, in the first step, transitions that do not result in a state satisfying $\varphi$. These transitions are only enabled if there is a token in $p_\varphi$. The net is shown in Figure 11.4.

The next lemma provides the essential step to use decision procedures for Petri nets in order to solve the model checking problem.

**Lemma 11.6.**

*(a)* $\mathfrak{M}, q_0 \models \langle\rho\rangle\Diamond\varphi$ *iff there is a run in $N^\Diamond$ on which there is a token in $p_\varphi$ at some moment where $N^\Diamond$ is the Petri net that equals $N_{\{q_0\},\rho}(\mathfrak{M}, Q_\mathfrak{M}, \varphi)$ with the exception that the initial token in $p_S$ is in $p_\varphi$ instead iff $q_0 \models^{prop} \varphi$.*

*(b)* $\mathfrak{M}, q_0 \models \langle\rho\rangle \bigcirc \varphi$ *iff there is a run in $N^{\bigcirc}_{\{q_0\},\rho}(\mathfrak{M}, \{q_0\}, \varphi)$ on which there is a token in $p_\varphi$ at some moment.*

*(c)* $\mathfrak{M}, q_0 \models \langle\rho\rangle\Box\varphi$ *iff there is a run in $N^\Box$ on which there never is a token in $p_{\neg\varphi}$ where $N^\Box$ is the Petri net that equals $N_{\{q_0\},\rho}(\mathfrak{M}, Q_\mathfrak{M}, \neg\varphi)$ with the exception that the initial token in $p_S$ is in $p_{\neg\varphi}$ instead iff $q_0 \not\models^{prop} \varphi$.*

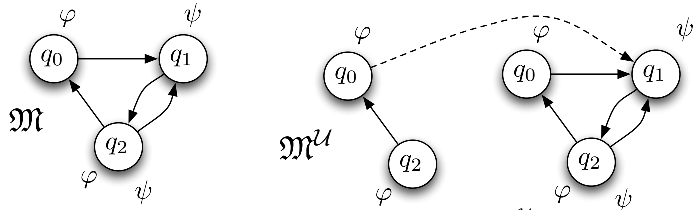



**Fig. 11.5.** Extending the RBM $\mathfrak{M}$ to $\mathfrak{M}^{\mathcal{U}}$ for $\varphi\mathcal{U}\psi$.

*Proof.* (a) Let $\lambda$ be a $\rho$-feasible path satisfying $\Diamond\varphi$ and let $i$ be the minimal index on $\lambda$ with $\lambda[i] \models \varphi$. By Lemma 11.4 there is a corresponding run in the PN. Particularly, there is a token in $p_S$ before the transition $t_i = t_{\lambda[i]\lambda[i+1]}$ fires and no token in $p_\varphi$ (apart from the case $q_0 \models^{\text{prop}} \varphi$). Thus, once $t_i$ has fired there is a token in $p_\varphi$. The other direction is proven analogously.

(b) The left to right direction is handled as in (a). For the other direction we observe that only transitions to $\varphi$-states are enabled at the beginning. If such a transition has fired all other transitions are enabled as well. Then, the claim follows with Lemma 11.4.

(c) Let $\lambda$ be a $\rho$-feasible path satisfying $\Box\varphi$. In the net a token can only be in $p_{\neg\varphi}$ if a transition yielding to a state satisfying $\neg\varphi$ is executed. As such a transition does never fire on the run corresponding to $\lambda$ there never is a transition in $p_{\neg\varphi}$ on this very run. Analogously, if there is a run such that there never is a token in $p_{\neg\varphi}$ this corresponds to a path containing no states not satisfying $\varphi$.

■

It remains to link the "until" case to Petri nets. For this, we consider the problem whether $\mathfrak{M}, q_0 \models \langle\rho\rangle\varphi\mathcal{U}\psi$. Let $\mathfrak{M}^\varphi$ be the restriction of $\mathfrak{M}$ to states in which $\varphi$ holds. Now, $\mathfrak{M}^{\mathcal{U}}$ is the model that glues together $\mathfrak{M}^\varphi$ with $\mathfrak{M}$ as follows: Each state $q$ in $\mathfrak{M}^\varphi$ is connected to a state $q' \in \mathfrak{M}$ if $q \rightarrow_{\mathfrak{M}} q'$ and $q'$ satisfies $\psi$. The construction is illustrated in Figure 11.5. States are relabelled if necessary.

**Lemma 11.7.** *Suppose $q_0 \models^{prop} \varphi$ (the other cases are trivially decidable). $\mathfrak{M}, q_0 \models \langle\rho\rangle\varphi\mathcal{U}\psi$ iff there is a run in $N^{\mathcal{U}}$ on which there is a token in $p_\psi$ at some moment where $N^{\mathcal{U}}$ is the Petri net that equals $N_{\{q_0\},\rho}(\mathfrak{M}^{\mathcal{U}}, Q_{\mathfrak{M}^{\mathcal{U}}}, \psi)$ with the exception that the initial token in $p_S$ is in $p_\psi$ instead iff $q_0 \models^{prop} \psi$.*

*Proof.* The construction ensures that only states satisfying $\varphi$ are visited until a state $\psi$ is visited. The rest follows from Lemma 11.6(a). ■

### 11.2.2 Decidability result

Finally, we show that the questions about Petri nets which were introduced in the previous two lemmata can be decided. Therefore, we reduce model checking to the extended reachability problem [Jančar, 1990] introduced in Definition 4.59; the latter was shown to be decidable (cf. Theorem 4.60). We have the following reductions.

**Lemma 11.8.** *Assume the same notation as in Lemma 11.6 and 11.7.*

*(a) There is a run in $N^{\bigcirc}$ on which there is a token in $p_\varphi$ at some moment iff $(N^{\bigcirc}, (\{p_\varphi\}, f_1))$ is in ExtReach where $f_1$ is the constant function 1.*
*(b) There is a run in $N^{\Diamond}$ on which there is a token in $p_\varphi$ at some moment iff $(N^{\Diamond}, (\{p_\varphi\}, f_1))$ is in ExtReach where $f_1$ is the constant function 1.*
*(c) There is a run in $N^{\Box}$ on which there never is a token in $p_{\neg\varphi}$ iff $(N^{\Box}, (\{p_{\neg\varphi}\}, f_0))$ is in ExtReach where $f_0$ is the constant function 0.*
*(d) There is a run in $N^{\mathcal{U}}$ on which there is a token in $p_\psi$ at some moment iff $(N^{\mathcal{U}}, (\{p_\psi\}, f_1))$ is in ExtReach where $f_1$ is the constant function 1.*

*Proof.* (a) From Proposition 11.5 we observe the following. There is a run on which there is a token in $p_\varphi$ at some moment iff there is a run on which there is a token in $p_\varphi$ infinitely often iff there is a run on which there is exactly one token in $p_\varphi$ infinitely often iff $(N^{\bigcirc}, (\{p_\varphi\}, f_1))$ is in *ExtReach.*

(b-d) These cases are handled analogously. ■

Thanks to Lemma 11.8 we obtain the following decidability result.

**Theorem 11.9.** *The model checking problem for* $\mathbf{RTL}_{\mathcal{R}es^\oplus}$ *over* RBMs *is decidable.*

*Proof.* Let $\langle\rho\rangle\gamma$ be a formula such that $\gamma$ does not contain any more path quantifiers. According to Lemma 11.6 we can reduce the model checking problem $\mathfrak{M}, q \models \langle\rho\rangle\gamma$ to a reachability problem over Petri nets. In turn, this problem can be reduced to the *ExtReach* problem (cf. Definition 4.59) by Lemma 11.8. The decidability of the model checking problem follows from the decidability of *ExtReach* (cf. Theorem 4.60). ■

*Remark 11.10.* We note that if a marking is reachable an appropriate sequence of transitions is constructed; this sequence can also be used to construct an appropriate $\kappa$ for the $\kappa$-model cover graph (cf. Section 8.2.1). One simply takes the maximum of all markings of all resource types along this sequence. If the state is not reachable, $\kappa$ is chosen arbitrarily (cf. Remark 8.30 and Example 8.18).

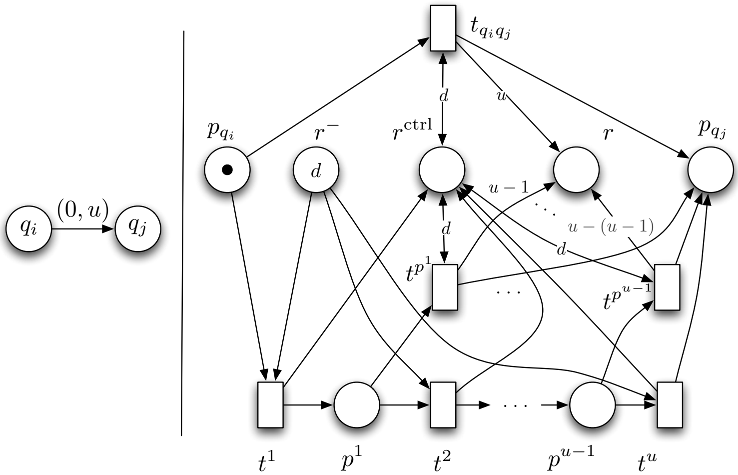



**Fig. 11.6.** Example of a PN construction for non-feasible resource sets: The left-hand RBM with a single resource $r$ with $\rho(r) = -d$ is converted to the right-hand PN.

We extend the previous construction to be able to deal with non-feasible resource sets and get the main result.

For non-feasible initial resource sets, we can still have a feasible path, in case no resources with negative amount are ever required in the run (resources can still be produced!).

We encode a non-feasible resource set by splitting each resource place $r$ of the Petri net into a place for a positive number of resources, $r$, and a place for a negative number of resources, $r^-$.

Further, we need to ensure in our net, that whenever resources are produced a positive number of tokens is placed on the positive resource place (only if no tokens are present in the negative resource place) or a number of tokens is removed from the negative resource place. Combinations are possible, if the number of resources produced is larger than the negative number of resources currently available. In the latter case all resources are removed from the negative resource place and the remaining difference is placed into the positive place. Therefore, we introduce a special resource control state, $r^{\mathrm{ctrl}}$, that "deactivates" the new part of the construction once a non-negative amount of resources is available.

In the following we will describe the construction in detail. Consider the transition of an RBM at the left-hand side of Figure 11.6. For simplicity, we only consider a single resource-type $r$. The transition consumes zero units of $r$ and produces $u$ units (note, that if the transition does also consume of this resource type we take the standard construction from Theorem 11.9).

Suppose, we would like to model check a formula $\langle\rho\rangle\gamma$ with $\rho(r) = -d$, that is, there is an initial debt of $d$ units of resource $r$. Firstly, we add a transition $t_{q_iq_j}$ from $p_{q_i}$ to $q_{q_j}$ which is only enabled if there are $d$ units in the resource control state $r^{\text{ctrl}}$ and a token on $p_{q_i}$. We add $u$ transitions $t^1, \ldots, t^u$; $u-1$ places $p^1, \ldots, p^{u-1}$; and $u-1$ intermediate transitions $t^{p^1}, \ldots, t^{p^{u-1}}$. Their connections are shown in the right-hand part of Figure 11.6. Each transition $t^i$ can only be enabled if there is a debt of resources (i.e. tokens in $r^-$). Such a transition takes one token from $r^-$ and moves it to the control state $r^{\text{ctrl}}$. Once, there are $d$ tokens in the control state the transitions $t^{p^i}$ can be enabled (while $t^i$ can no longer be enabled) and the remaining produced resources are added to the resource place $r$. The net has the following properties.

**Proposition 11.11.**

1. *There are $x$ tokens in $r^-$ iff there are $d-x$ token in $r^{ctrl}$ for $x \in \{0, \ldots, d\}$. (That is, $r^-$ and $r^{ctrl}$ are complementary places.)*
2. *Transitions $t_{q_iq_j}$ and $t^{p^1}, \ldots, t^{p^{u-1}}$ can only fire if there are $d$ tokens in $r^{ctrl}$.*
3. *The number of tokens in $r^{ctrl}$ is bounded by $d$ and it is monotonically increasing.*
4. *The number of tokens in $r^-$ is monotonically decreasing.*
5. *If there is a token in place $p_{q_i}$ and there are $d$ tokens in $r^{ctrl}$ only the transition $t_{q_iq_j}$ is enabled.*
6. *There can only be tokens in $r$ if there are no tokens in $r^-$.*

*Proof.*

1. Firstly, observe that for any transition $t^i$ a token is removed from $r^-$ and added to $r^{\text{ctrl}}$ and there is no other way how resources can be consumed form $r^-$ and all other transition producing/consuming from $r^{\text{ctrl}}$ in turn consume/produce the same amount.
2. Obvious.
3. Obvious as only the transitions $t^i$ can add resources; for all other transitions the number of tokens remains constant.
4. Obvious.
5. In this case, the firing of $t_{q_iq_j}$ only depends on whether there is a token in $p_{q_i}$. Moreover, there are no tokens in $r^-$; hence, all transitions $t^i$ are disabled.

6. Tokens can only be added to $r$ if transitions $t^{p^i}$ or transition $t_{q_i q_j}$ fire. These transitions are only enabled if there are $d$ tokens in $r^{\mathrm{ctrl}}$ but then there are no tokens in $r^-$.

■

The next lemma shows that the net works as intended. The result follows from the previous proposition.

**Lemma 11.12.**

1. *Let there be a token in $p_{q_i}$, $d' \leq d$ tokens in $r^-$, $d - d'$ tokens in $r^{ctrl}$, and no tokens in $r$. Let $\sigma$ be the minimal length firing sequence such that there is a token in $p_{q_j}$. Then, after executing $\sigma$ there are* $\max\{0, d' - u\}$ *tokens in $r^-$,* $\min\{d, d - d' + u\}$ *tokens in $r^{ctrl}$, and* $\max\{0, u - d'\}$ *tokens in $r$.*
2. *On the other hand, if there is a token in $p_{q_i}$, $d$ tokens in $r^{ctrl}$, zero tokens in $r^-$ and $k$ tokens in $r$ then, after executing $\sigma$ there are $k + u$ tokens in $r$, $d$ tokens in $r^{ctrl}$, and zero tokens in $r^-$.*

*Proof.* Assume the conditions of the lemma are satisfied.

1. We consider the possible cases for $d'$.
   $d' = d$. Firstly, let us assume that $d' = d$. Then, there are no tokens in $r^{\mathrm{ctrl}}$. Only transition $t^1$ can fir, a token is added to $r^{\mathrm{ctrl}}$ and removed from $r^-$. The transitions $t^1, \ldots, t^m$ fire for the next $m := \min\{d, u\}$ steps. Suppose $d < u$. Once place $p^d$ is reached, there are $d$ tokens in $r^{\mathrm{ctrl}}$ and 0 tokens in $r^-$. Transition $t^{p^d}$ fires and $u - d = u - d'$ tokens are added to $r$, one token to $p_{q_j}$, $0 = \max\{0, d' - u\}$ tokens are in $r^-$ and $d = \min\{d, u\}$ tokens are in $r^{\mathrm{ctrl}}$.
   Now suppose $d \geq u$. Then $t^1, \ldots t^u$ fire and there is a token in $p_{q_j}$, $d - u = \max\{0, d' - u\}$ tokens in $r^-$, $u = \min\{d, u\}$ tokens in $r^{\mathrm{ctrl}}$, and $0 = \max\{0, u - d'\}$ tokens in $r$.
   $d' = 0$. In this case there are $d$ tokens in $r^{\mathrm{ctrl}}$. Transition $t_{q_i q_j}$ fires and there are $u$ tokens in $r$ and one token in $p_{q_j}$ as stated in the lemma.
   $0 < d' < d$. There are $d - d'$ tokens in $r^{\mathrm{ctrl}}$. Then, transitions $t^1, \ldots t^m$ with $m = \min\{d', u\}$ fire. Let $d' < u$. Once place $p^{d'}$ is reached, there are $d = d - d' + d'$ tokens in $r^{\mathrm{ctrl}}$ and 0 tokens in $r^-$. Transition $t^{p^{d'}}$ fires and $u - d'$ tokens are added to $r$, one token to $p_{q_j}$, $0 = \max\{0, d' - u\}$ tokens are in $r^-$ and $d = \min\{d, d - d' + u\}$ tokens are in $r^{\mathrm{ctrl}}$.
   Now suppose $d' \geq u$. Then $t^1, \ldots t^u$ fire and there is a token in $p_{q_j}$, $d' - u = \max\{0, d' - u\}$ tokens in $r^-$, $d - d' + u = \min\{d, d - d' + u\}$ tokens in $r^{\mathrm{ctrl}}$, and $0 = \max\{0, u - d'\}$ tokens in $r$.
2. Let there be a token in $p_{q_i}$, $d$ tokens in $r^{\mathrm{ctrl}}$, zero tokens in $r^-$ and $k$ tokens in $r$. Then, $t_{q_i q_j}$ fires and there are $k + u$ tokens in $r$ and a token in $p_{q_j}$. The number of tokens in $r^-$ and $r^{\mathrm{ctrl}}$ remains the same.

■

We combine the construction of Section 11.2.1 with the extended construction sketched above. If there is an initial debt of a resource type $r$ and the net consumes from this resource, the encoding of a transition shown in Figure 11.1 is replaced by the one given in Figure 11.6. Hence, the setting of non-feasible resource sets reduces to the feasible one. Thanks to Theorem 11.9 we obtain the following result:

**Theorem 11.13 (Model Checking RTL: Decidability).** *The model checking problem for* **RTL** *over* RBMs *is decidable.*

## 11.3 Resource-Bounded Agent Logic

In this section we analyse the model checking problem and consider how the variously restricted settings influence its complexity. It is well known that the model checking problems for $\mathbf{ATL_{IR}}$, $\mathbf{ATL^*_{IR}}$, and $\mathbf{ATL^*_{IR}}$ are **P**-complete, **PSPACE**-, and **2EXPTIME**-complete, respectively (cf. Section 5.3). Model checking **RTL** has been shown decidable in [Bulling and Farwer, 2010a] (cf. Section 11.2), and the same holds for **RBCL** [Alechina et al., 2009b]. Here, we show that the latter two cases form an exception; the general resource-bounded settings turn out to be undecidable due to the possibility of *producing* resources.

### 11.3.1 Decidability Results

For both bounded settings introduced in Section 8.4.4 we have that along each resource extended path there are only finitely many reachable states from $Q \times \mathsf{En}$. Hence, given an endowment, we can 'unravel' a given RBAM and apply standard $\mathbf{ATL}^*$ model checking [Alur et al., 2002] which is proven to be decidable. The unraveling however may yield finite paths (i.e. states with no successor) and requires an (straightforward) extension of existing algorithms.

**Proposition 11.14.** *It is decidable to determine whether a given* RBAM *is k-bounded for* $\eta$.

*Proof.* We apply the cover graph construction for RBMs presented in Definition 8.13. That is, we build a new model with states drawn from $Q \times \mathsf{En}$. Let $q^I \in Q$. Then, we "unravel" the model $\mathfrak{M}$ from each $q^I$ on keeping track of the resources in the states $(q, \eta')$. Once we encounter a new state $(q', \eta_2)$ and did already create a state $(q, \eta_1)$ with $q' = q$ and $\eta_2(a,r) \geq \eta_1(a,r)$ for all agents and resources we do not add $(q', \eta_2)$ but rather add the state $(q', \eta_\omega)$ with $\eta_\omega(a,r) = \omega$ for which $\eta_2(a,r) > \eta_1(a,r)$ . We use $\omega$ to denote that we can create any bounded amount of resources. This construction eventually converges to a finite structure (also cf. the proof of Theorem 8.20 for details).

Finally, the model is $k$ bounded if in each unraveling for each state $q \in Q$ there is no state $(q', \eta')$ such that there is an agent $a$ and a resource type $r$ with $\eta(a, r) > k$ or $\eta(a, r) = \omega$. ∎

**Theorem 11.15.** *Model checking* $\mathbf{RAL}^*_R$ *(and all other variants discussed here) is decidable over the class of bounded* RBAMs*.*

*Proof.* For a given endowment we 'unravel' the RBAM such that the states are given by $(q, \eta)$ where $q$ is a state of the original model and $\eta$ an endowment (cf. the proof of Proposition 11.14). As in a $k$-bounded model only finitely many resource-quantities can occur (and the set $Q$ is finite), there are only finitely many such state/ endowment combinations. Hence, the unraveling converges at some moment (in comparison to the proof of Proposition 11.14 no $\omega$-resource quantities have to be introduced). Then, we interpret the $\mathcal{L}_{RAL^*}$-formula as $\mathcal{L}_{ATL^*}$-formula and model check it in the resulting CGS. Formulae are evaluated bottom-up. Since each state is coupled with an endowment also non-resource-flat formulae can be treated in such a way. We would like to mention that this procedure is extremely costly. ∎

Following the same line of reasoning we can prove the next result which is of practical importance as it allows to obtain a decidable model checking result for all logics and *all* RBAMs.

**Theorem 11.16.** *The model checking problem for* $\mathbf{RAL}^*_R$ *(and all other variants discussed here) over the $k$-bounded semantics is decidable for any $k \in \mathbb{N}$.*

### 11.3.2 Two Counter Automata.

The undecidability proofs are done by simulating a *two counter automaton (*TCA*)* $\mathcal{A}$ (cf. [Hopcroft and Ullman, 1979]) and a reduction to the *halting problem on empty input* (we write $\mathcal{A}\downarrow$ for '$\mathcal{A}$ halts on empty input'). A TCA is essentially a (nondeterministic) push-down automaton with two stacks and exactly two stack symbols (one of them is the initial stack symbol). This kind of machine has the same computation power as a Turing machine.

**Definition 11.17 (Two-counter automaton (cf. [Hopcroft and Ullman, 1979])).** *A* TCA $\mathcal{A}$ *is given by*

$$(S, \Gamma, s^{init}, S_f, \Delta)$$

*where $S$ is a finite set of states, $\Gamma$ is the finite input alphabet, $s^{init} \in S$ is the initial state, $S_f \subseteq S$ is the set of final states, and $\Delta \subseteq (S \times \Gamma \times \{0,1\}^2) \times (S \times$*

$\{-1,1\}^2)$ *is the transition relation such that if* $((s,a,E_1,E_2),(s',C_1,C_2)) \in \Delta$ *and* $E_i = 0$ *then* $C_i \neq -1$ *for* $i = 1,2$ *(to ensure that an empty counter cannot further be decremented). In the case of an empty input, we ignore the alphabet and assume* $\Delta \subseteq (S \times \{0,1\}^2) \times (S \times \{-1,1\}^2)$.

A TCA can be considered as a transition system equipped with two counters that influence the transitions. Each transition step of the automaton may rely on any of the counters being zero or non-zero and in each step the counters can be incremented or decremented. It is important to stress that a TCA can only distinguish between a counter being zero or non-zero. Consider the transition $((s,E_1,E_2),(s',C_1,C_2)) \in \Delta$. Here, $E_i = 1$ (resp. $= 0$) represents that counter $i$ is non-empty (resp. empty) and $C_k = 1$ (resp. $= -1$) denotes that counter $i$ is incremented (resp. decremented) by 1. The transition encodes that in state $s$ the automaton can change its state to $s'$ provided that the first (resp. second) counter meets condition $E_1$ (resp. $E_2$). The value of counter $k$ changes according to $C_k$ for $k = 1,2$. The transition $((s,1,0),(s',-1,1) \in \Delta$, for example, is enabled if the current sate is $s$, counter 1 is non-empty, and counter 2 is empty. If the transition is selected the state changes to $s'$, counter 1 is decremented and counter 2 is incremented by 1.

The general mode of operation is as for pushdown automata. In particular, a *configuration* is a triple $(s,v_1,v_2) \in S \times \mathbb{N}_0^2$ describing the current state ($s$), the value of counter 1 ($v_1$) and of counter 2 ($v_2$). A *computation* $\delta$ is a sequence of subsequent configurations that can emerge by transitions according to $\Delta$ such that the first state is $s^{\text{init}}$. An *accepting* configuration is a finite computation $\delta = (s_i, v_1^i, v_2^i)_{i=1,\dots,k}$ where the last state $s_k \in S_f$, i.e., it is a final state. We use $\delta_i = ((s_i,E_1^i,E_2^i),(s_{i+1},C_1^i,C_2^i)) \in \Delta$ to denote the tuple that leads from the $i$th configuration $(s_i,v_1^i,v_2^i)$ to the $i+1$st configuration $(s_{i+1},v_1^{i+1},v_2^{i+1})$ for $i < k$. In particular, we have that $v_j^{i+1} = v_j^i + C_j^i$ for $j = 1,2$.

### 11.3.3 Idea of the Reductions

In order to show that model checking of resource-bounded agent logics is undecidable, we reduce the halting problem to these logics. The specific construction varies for each logic. In the following we present the general idea. Detailed proofs can be found in Appendix B.7. Let $\mathcal{A} = (S, \Gamma, s^{\text{init}}, S_f, \Delta)$ be a TCA. We represent the value of the two counters as resource types $R_1$ and $R_2$, respectively. For each state of the automaton, we add a state to the model and we label the accepting states in $S_f$ by a proposition $\mathsf{halt}$. The increment and decrement of counter values are modelled by actions producing and consuming from the corresponding resource type. The general idea underlying all the reductions is as follows (the path formula depends on the specific logic $\mathbf{L}$ considered):

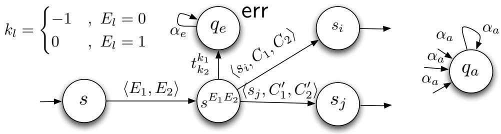



**Fig. 11.7.** Transformation of transitions $(s, E_1, E_2)\Delta(s_i, C_1, C_2)$ and $(s, E_1, E_2)\Delta(s_j, C'_1, C'_2)$ .

($\star$) *$\mathcal{A}\!\downarrow$ iff there is a path in the* RBAM *along which a path formula $\gamma_{\mathsf{L}}$ is true.*

The satisfying path in the RBAM corresponds to an accepting computation of the automaton. The general mode of operation is straightforward and only the following difficulty remains: It is *not* possible to test whether a counter (i.e. a resource type) is empty in *any* of the resource-bounded agent logics. This causes difficulties in the reductions. For example, consider a tuple $((s, 1, 0), (s', -1, 1)) \in \Delta$. It can only be chosen if the second counter is actually empty. But, because we cannot directly test whether a resource type is empty, we need to introduce a workaround. This is the sophisticated part in the reductions (sometimes easier sometimes harder, depending on the expressiveness of the used logic). The encoding of a transition $r := ((s, E_1, E_2), (s', C_1, C_2))$ is a three-step process (cf. Figure 11.7). In a state $s$ of the RBAM (we are economic and use the same notation for elements of the model and the automaton) an agent performs an action $\langle E_1, E_2 \rangle$ in order to 'select' $r$. This results in a 'test' state $s^{E_1 E_2}$. In this state, an action $\langle s', C_1, C_2 \rangle$ with resource-costs corresponding to the values of $C_i$ can be executed (i.e. the action produces/consumes $C_i$ resources of $R_i$). Clearly, such an action is only successful if sufficient resources are available. The check whether a counter/resource type is empty or not, takes place at the intermediate state $s^{E_1 E_2}$. In these states, a *non-cost-free* action $t^{k_1}_{k_2}$ for $k_i \in \{0, -1, 1\}$ leading to an 'error state' $q_e$ is available. Thus, if a counter should be zero according to the transition $t$; then, such a test action must not be performable. Hence, ($\star$) can be refined to the following:

($\star\star$) *$\mathcal{A}\!\downarrow$ iff there is a path in the* RBAM *such that eventually* halt *and along which there is no way to reach the error state $q_e$.*

Intuitively, if the error state cannot be reached along a path the selection of transitions is valid in the sense described above (i.e. it corresponds to an accepting computation of the automaton).

### 11.3.4 Undecidability: Non-Flat Languages.

We begin with specialised settings for non-flat languages. In the case of $\mathsf{RAL}_r$ we test whether there is a path such that eventually $\mathsf{halt}$ and in no state a transition to $\mathsf{err}$ is possible. In order to test whether the error state can be reached we make use of the non-resource-flatness of the logic. Formally, we show:

$$\mathcal{A}\downarrow \text{ iff } \mathfrak{M}^{\mathcal{A}}, s^{\text{init}}, \eta_0 \models_r \neg\langle\langle\emptyset\rangle\rangle^{\eta_0}_{\mathbb{A}\mathrm{gt}}\neg((\neg\langle\langle\emptyset\rangle\rangle \bigcirc \neg\mathsf{err})\mathcal{U}\mathsf{halt}).$$

The endowment $\eta_0$ equips agents with no resources.

**Theorem 11.18.** *Model checking* $\mathsf{RAL}_r$ *is undecidable, even in the single agent case; hence also,* $\mathsf{RAL}_r^+$ *and* $\mathsf{RAL}_r^*$ *are undecidable.*

The complete proof is given on page 325.

In the previous case it was essential to keep track of the resources of the opponent. Here, we show that also the proponent-restricted setting is undecidable if we allow perfect recall strategies. A perfect recall strategy of the *proponent* is used to *encode* the computation of the automaton. Similar to Theorem 11.18, we obtain the following reduction:

$$\mathcal{A}\downarrow \text{ iff } \mathfrak{M}^{\mathcal{A}}, s^{\text{init}}, \eta_0 \models_R \langle\langle 1\rangle\rangle^{\eta_0}((\neg\langle\langle 1\rangle\rangle \bigcirc \mathsf{err})\mathcal{U}\mathsf{halt}).$$

**Theorem 11.19.** *Model checking pr-*$\mathsf{RAL}_R$ *(even without the release operator) is undecidable in the single-agent case; hence, also pr-*$\mathsf{RAL}_R^+$*, pr-*$\mathsf{RAL}_R^*$*,*$\mathsf{RAL}_R$*,* $\mathsf{RAL}_R^+$*, and* $\mathsf{RAL}_R^*$ *are undecidable.*

The complete proof is given on page 326.

For the next setting, the proponent has once again no memory available. In turn, an additional agent (opponent agent 2) is used to model the computation (as in Theorem 11.18) and the proponent (agent 1) keeps track of the resources (as in Theorem 11.19). It is important to note that the language is *not* resource-flat. The idea of the construction is shown in Figure 11.8. Then, we have that

$$\mathcal{A}\downarrow \text{ iff } \mathfrak{M}^{\mathcal{A}}, s^{\text{init}}, \eta_0 \models_r \neg\langle\langle 1\rangle\rangle^{\eta_0}\neg((\neg\langle\langle 2\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \bigcirc \mathsf{err})\mathcal{U}\mathsf{halt}).$$

**Theorem 11.20.** *Model checking pr-*$\mathsf{RAL}_r$ *is undecidable for models with at least two agents; hence, also pr-*$\mathsf{RAL}_r^+$ *and pr-*$\mathsf{RAL}_r^*$ *are undecidable.*

The complete proof is given on page 327.

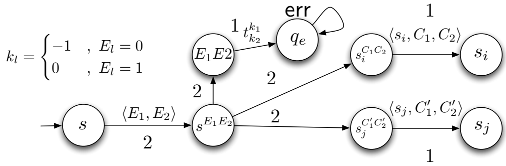


**Fig. 11.8.** Construction used in the proof of Theorem 11.20 for $(s, E_1, E_2)\Delta(s_i, C_1, C_2)$ and $(s, E_1, E_2)\Delta(s_j, C_1', C_2')$.

### 11.3.5 Undecidability: Resource-Flat Languages

Resource-flat logics seem less difficult to verify. In the reduction it is not possible to have nested operators in order to verify whether the resources in a state are actually zero (compare the techniques introduced in the above); more precisely, in the formula $\langle\!\langle 1\rangle\!\rangle^{\eta_0}((\neg\langle\!\langle 1\rangle\!\rangle \bigcirc \mathsf{err})\mathcal{U}\mathsf{halt})$ the test whether the error state is reachable modelled by the second cooperation modality used the resources available at that very moment. Such scenarios cannot be modelled with resource-flat languages.

We show that the perfect recall scenario and two agents can be used to 'overcome this limitation'. The proponent (agent 1) is used to simulate the computation of the automaton where the opponent (agent 2) tries to enter the error state in each test state; hence, no nested cooperation modality is needed. The setting is similar to the one shown in Figure 11.7 extended with a second agent. We show:

$$\mathcal{A}\!\downarrow \text{ iff } \mathfrak{M}^{\mathcal{A}}, s^{\text{init}}, \eta_0 \models_R \langle\!\langle 1\rangle\!\rangle^{\eta_0}_{\mathbb{A}\mathrm{gt}} \Diamond \mathsf{halt}.$$

**Theorem 11.21.** *Model Checking rf-* $\mathbf{RAL}_R$ *is undecidable for models with at least two agents; thus, also rf-* $\mathbf{RAL}^+_R$ *and rf-* $\mathbf{RAL}^*_r$ *are undecidable.*

The complete proof is given on page 328.

At present, the decidability of the resource-flat *and* proponent-restricted versions of $\mathcal{L}_{RAL^+}$ and $\mathcal{L}_{RAL}$ with the standard semantics are open. However, by using the apparently stronger infinity-semantics ($\models^\infty_R$) we can prove the undecidability of *rf-pr-*$\mathcal{L}_{RAL}$ and thus also of *rf-pr-*$\mathbf{RAL}^*_R$ by Proposition 8.54. We do this by showing

$$\mathcal{A}\!\downarrow \text{ iff } \mathfrak{M}^{\mathcal{A}}, s^{\text{init}}, \eta_0 \models^\infty_R \langle\!\langle 1\rangle\!\rangle^{\eta_0}(\neg\mathsf{err})\mathcal{U}\mathsf{halt}.$$

The construction is sketched in Figure 11.9. Essentially, the opponent (agent 2) may decide to enter the 'test *loop*' in $s^{E_1E_2}$. This 'bad' *loop* can only be avoided if agent 1 chooses good transitions of the automaton. Finite dead-end paths are disregarded thanks to the infinity-semantics.

**Theorem 11.22.** *Model Checking rf-pr-*$\mathbf{RAL}^*_R$*, rf-pr-*$(\mathcal{L}_{RAL}, \models^\infty_R)$*, and rf-pr-*$(\mathcal{L}_{RAL}, \models^\infty_R)$ *is undecidable for models with at least two agents.*

The complete proof is given on page 329.

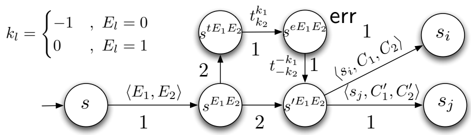



**Fig. 11.9.** Construction used in the proof of Theorem 11.22 for $(s, E_1, E_2)\Delta(s_i, C_1, C_2)$ and $(s, E_1, E_2)\Delta(s_j, C'_1, C'_2)$.

| | $\mathcal{L}_{RAL^*}$ | $\mathcal{L}_{RAL^+}$ | $\mathcal{L}_{RAL}$ | $pr$-$\mathcal{L}_{RAL^*}$ | $pr$-$\mathcal{L}_{RAL^+}$ | $pr$-$\mathcal{L}_{RAL}$ |
|---|---|---|---|---|---|---|
| $\models_R$ | $U^1$ | $U^1$ | $U^1$ | $U^1$ | $U^1$ | $U^1$ |
| $\models_r$ | $U^1$ | $U^1$ | $U^1$ | $U^2$ | $U^2$ | $U^2$ |
| $rf + \models_R$ / $\models_R^\infty$ | $U^2$ | $U^2$ | $U^2$ | $U^2$/ $U^2_\infty$ | ? / $U^2_\infty$ | ? / $U^2_\infty$ |
| $rf + \models_r$ | ? | ? | ? | ? | ? | ? |
| $\models_R^k$, $\models_r^k$ | **D** | **D** | **D** | **D** | **D** | **D** |

**Table 11.1.** Overview of model checking decidability results. Each cell represents the logic over the language given in the column using the semantics given in the row. The content of each cell indicates whether the model checking problem is decidable (D) or undecidable ($U^x$). $x$ indicates the number of required agents. $U^2_\infty$ refers to the semantics $\models_R^\infty$.

### 11.3.6 Overview of the Results

Our analysis, summarised in Table 11.1, shows that the combination of various settings and languages influences the difficulty of the model checking problem. Although we do not claim that our results with respect to the number of agents are optimal they show an interesting pattern. One can often compensate a lack of expressiveness caused by various restrictions on the language or semantics by taking more agents into account. The most difficult cases seem to be the ones using the perfect recall semantics. Resource-flatness suggests to be important for decidable fragments, particularly in combination with memoryless strategies.

The question for the resource-flat proponent-restricted languages $\mathcal{L}_{RAL^+}$ and $\mathcal{L}_{RAL}$ under the $R$-semantics is *still open*, while the case is proven undecidable if only infinite paths are considered. Also open is the case of resource-flat languages over the $r$-semantics. The two bounded settings are shown to be decidable.

Finally, we would like to mention that the result from Section 11.2 on the decidability of **RTL** matches the results presented here, since it corresponds to the single-agent case of $rf$-$pr$-$\mathbf{RAL}_R$.

## 11.4 Summary

Our main objective of this chapter has been the analysis whether it is possible to verify resource-bounded agents in diverse settings. We have shown that the single-agent case is decidable for **RTL** and also for **RTL**$^{\star}$ under various restrictions. We are particularly interested in finding constraints that would make the extended logic's model checking problem *efficiently* decidable for a relevant class of MASs.

Moreover, we have also addressed the multi-agent case and have shown undecidability for many fragments and identified the number of agents needed. We believe that these results are important and interesting for future work on strategic abilities under limited resources. Our results have shown that small changes in the language and semantics may influence whether model checking becomes decidable or undecidable (cf. for instance, the $\models_r^\infty$ and $\models_r$ semantics over $rf\text{-}pr\text{-}\mathcal{L}_{RAL}$). We have also considered bounded settings with decidable model checking problems.

# Part IV

# Conclusions

# 12

# Conclusions

## Contents



In this final chapter we summarise related work, present a summary, and give a brief outlook on directions for future work.

## 12.1 Related Work: A Summary

We summarise the discussion of related work that has been given throughout the thesis.

### 12.1.1 Strategic Ability and Knowledge

In this thesis we have proposed several extensions based on the linear time logic **LTL** [Pnueli, 1977], the computation-tree logics **CTL** and **CTL**$^*$ [Emerson and Halpern, 1986], and various variants of the alternating time temporal logics **ATL** and **ATL**$^*$ [Alur et al., 2002, 1997, 1998b] to which we also count coalition logic [Pauly, 2002]. The focus has been on the latter class of strategic logics.

In the last years, ATLs have also attracted a lot of attention by other researcher. In [Goranko and Jamroga, 2004] it was shown that CGSs and alternating transition systems provide an equivalent semantics, and that **CL** can be seen as the *next-time fragment* of **ATL**. Also the expressivity of various fragments have been considered, it was shown that if the "release" or "weak until" operator is added to the language of $\mathcal{L}_{ATL}$ it has the same expressive power as the syntactically more general logic $\mathbf{ATL}^+$ [Bulling and Jamroga, 2010a; Laroussinie et al., 2008; Harding et al., 2002]) but the latter enables a more succinct encoding of properties (this follows from the results in [Wilke, 1999]). In addition to that, standard issues about logics have been addressed. In [Goranko and van Drimmelen, 2003] a complete *axiomatisation* for $\mathbf{ATL}_{\mathbf{IR}}$ was presented. The *satisfiability problems* of $\mathbf{ATL}_{\mathbf{IR}}$ and $\mathbf{ATL}^*_{\mathbf{IR}}$ have also been considered by researchers: The problem was proven **EXPTIME**-complete for $\mathbf{ATL}_{\mathbf{IR}}$ [van Drimmelen, 2003; Walther et al., 2006] and **2EXPTIME**-complete for $\mathbf{ATL}^*_{\mathbf{IR}}$ [Schewe, 2008]. Axiomatisation and satisfiability of other variants of alternating time temporal logics still remain open, to the best of our knowledge.

On top of strategic ability, which is the focal point of ATLs, several extensions capable of dealing with imperfect information have been proposed. A first variant for incomplete information has been presented in [Alur et al., 2002]. Incomplete information has directly been included in the cooperation modalities similar to [Schobbens, 2004]; no explicit knowledge operators were introduced. In the presence of knowledge operators the interplay between strategic ability and knowledge has turned out very interesting and non trivial. The *alternating time temporal epistemic logic* **ATEL** from [van der Hoek and Wooldridge, 2003] extends **ATL** by standard epistemic concepts. The logics *feasible* **ATEL** [Jonker, 2003], *uniform* **ATEL** [Jamroga, 2003] and *alternating time temporal observational logic* [Jamroga and van der Hoek, 2004] are of the same kind and overcome some problems encountered with **ATEL**. Similarly, *epistemic temporal strategic logic* [van Otterloo and Jonker, 2004] restricts to undominated strategies. Finally, the logic **CSL** (*constructive strategic logic*) has been proposed in [Jamroga and Ågotnes, 2006, 2007], a very expressive logic that combines strategic ability and epistemic concepts in a neat way. The latter comes for the cost of a non-standard semantics.

Related to the interplay of strategic ability and knowledge is also the work presented in [Broersen et al., 2006]; there, it was shown how cooperation modalities can be decomposed into two parts in the context of **STIT** logic. A similar decomposition is considered in [Jamroga, 2008b] for the analysis of stochastic multi-agent systems.

### 12.1.2 Strategic Ability, Game Theory, and Rationality

ATLs do have a close relationship to basic game-theoretic concepts due to their semantics which is given in terms of winning strategies. It has been shown that the semantical models of ATLs, concurrent game structures (CGSs), have a close relationship to strategic and extensive form games known from game theory. The correspondence between extensive form games and the semantical models of **ATL** has been examined in [Jamroga et al., 2005] and was inspired by [Baltag, 2002; van der Hoek et al., 2005a]. It seems reasonable to refine **ATL** in such a way that it takes into account only "sensible" behaviour of agents. Two logics that can be used to implement these ideas from game theory directly are *game logic with preferences* **GLP** [van der Hoek et al., 2004] **ATLI** ("**ATL** with Intentions") [Jamroga et al., 2005]. The latter has served as a motivation for our logic **ATLP** defined in Section 6.1.

More generally, there seem to be two focal points in this context. Research within game theory understandably favours work on the *characterisation* of various types of rationality (and defining most appropriate solution concepts). Applications of game theory, also understandably, tend toward *using* the solution concepts in order to predict the outcome in a given game (in other words, to "solve" the game).

The first issue has been studied in the framework of logic, for example in [Bacharach, 1987; Bonanno, 1991; Stalnaker, 1994, 1996]; more recently, game-theoretical solution concepts have been characterised in dynamic logic [Harrenstein et al., 2002, 2003], dynamic epistemic logic [Baltag, 2002; van Benthem, 2003], and extensions of **ATL** [van der Hoek et al., 2005a; Jamroga et al., 2005].

The second thread seems to have been neglected in logic-based research: The work [van Otterloo et al., 2004; van der Hoek et al., 2004; van Otterloo and Roy, 2005; van Otterloo and Jonker, 2004] are the only exceptions we know of. Moreover, every proposal from [van Otterloo et al., 2004; van der Hoek et al., 2004; van Otterloo and Roy, 2005; van Otterloo and Jonker, 2004] commits to a particular view of rationality (Nash equilibria, undominated strategies etc.).

Game logic from [Parikh, 1985] is another logic about games. It builds upon propositional dynamic logic (PDL) [Fischer and Ladner, 1979]. The logic can be used to reason about determined two player games [Blackburn et al., 2006]. The idea is to interpret PDL operators in a game theoretic context and to add some new constructs. Several other logic-related methods have been proposed; e.g., in [van Benthem, 2003] and [Bonanno, 2002].

Another way to incorporate more sophisticated game-theoretic concepts is to modify the strong semantics of ATLs that share the "all-or-nothing" attitude of many logical approaches to computation, justified by von Neumann's maximin evaluation of strategies in classical game theory [von Neumann and

Morgenstern, 1944]. The logic **ATL** with probabilistic success which has been proposed in Section 7.4 softens this rigorous approach to success. Related work in this context is presented in [Jamroga, 2008a] and [de Alfaro et al., 2004].

### 12.1.3 Adding Rationality Concepts to Strategic Logics

In this thesis we have introduced **ATLP**, a logic which can be used to reason about temporal properties of rational play. The logic has been shown to be quite expressive. We have proposed embeddings of some related logics, e.g. of game logic with preferences [van Otterloo et al., 2004; van der Hoek et al., 2004] (Section 3.2.4) which allows to reason about what can happen under *particular* game-theoretical rationality assumptions. We observe that our framework is semantically similar to the approach of *social laws* [Shoham and Tennenholz, 1992; Moses and Tennenholz, 1995; van der Hoek et al., 2005b]. However, we refer to *strategy profiles* as rational or not, while social laws define constraints on agents' *individual actions*. Also, our motivation is different: In our framework, agents are expected to behave in a specified way because it is rational in some sense; social laws prescribe behaviour sanctioned by social norms and legal regulations.

In contrast to [Friedman and Halpern, 1994; Su et al., 2005; Bulling and Jamroga, 2006], the concept of plausibility presented in this article is *objective*, i.e. it does not vary from agent to agent. This is very much in the spirit of game theory, where rationality criteria are used in an analogous way. Moreover, it is *global*, because plausibility sets do not depend on the state of the system. We note, however, that the denotation of plausibility terms depends on the actual state.

The imperfect information variant of **ATLP**, *Constructive Strategic Logic with Plausibility* (**CSLP**), allows us to neatly define the relationship between epistemic and doxastic concepts, in a similar way as the logic **CTLKP** from [Bulling and Jamroga, 2007a]. This logic is a result of extending **CTLK** [Penczek and Lomuscio, 2003] by plausibility operators. In **CTLKP** plausibility assumptions were defined in terms of paths in the underlying system. Then, an agent's belief is given by its knowledge if only plausible paths were considered. The idea to build beliefs on top of plausibility has been inspired by [Su et al., 2005; Friedman and Halpern, 1994]. Another source of inspiration is [van der Hoek et al., 2004; van Otterloo and Jonker, 2004], where the semantics of ability was influenced by particular notions of rationality.

We have also considered how rational coalitions may form. For this purpose, we have proposed the logic **CoalATL**. We have followed an approach based on [Amgoud, 2005a,b], where an argumentation framework for generating coalition structures is defined. The approach is a generalisation of the framework of Dung for argumentation [Dung, 1995], extended with a *preference relation*. Previous research by Hattori *et al.* [Hattori et al., 2001] has also addressed the problem of argument-based coalition formation, but from a different perspective than ours. In [Hattori et al., 2001] the authors propose an argumentation-based negotiation method for coalition formation which combines a logical framework and an argument evaluation mechanism. The resulting system involves several user agents and a mediator agent. During the negotiation, the mediator agent encourages appropriate user agents to join a coalition in order to facilitate reaching an agreement. User agents advance proposals using a part of the user's valuations in order to reflect the user's preferences in an agreement. This approach differs greatly from our proposal, as we are not concerned with the negotiation process among agents, and our focus is on modelling coalitions within an extension of a highly expressive strategic logic, where coalition formation is part of the logical language.

Modelling argument-based reasoning with bounded rationality has also been the focus of previous research. In [Rovatsos et al., 2005] the authors propose the use of a framework for argument-based negotiation, which allows for a strategic and adaptive communication to achieve private goals within the limits of bounded rationality in open argumentation communities. In contrast with our approach, the focus here is not on extending a particular logic for reasoning about coalitions. Recent research on formalising coalition formation has been oriented towards adding more expressivity to Pauly's coalition logic [Pauly, 2002]. E.g. in [Ågotnes et al., 2007b], the authors define *quantified coalition logic*, extending coalition logic with a limited but useful form of quantification to express properties such as *"there exists a coalition $C$ satisfying property $P$ such that $C$ can achieve $\varphi$"*. In [Borgo, 2007], a semantic translation from coalition logic to a fragment of an action logic is defined, connecting the notions of coalition power and the actions of the agents. However, in none of these cases argumentation is used to model the notion of coalition formation as done in this thesis.

Also related is the work [Prakken and Vreeswijk, 2002; Chesñevar et al., 2000] in which argumentation frameworks have evolved as a successful approach to formalise common sense reasoning and decision making in multiagent systems (MASs). Application areas include issues such as joint deliberation, persuasion, negotiation, knowledge distribution and conflict resolution (e.g. [Tang and Parsons, 2005; Rahwan and Amgoud, 2006; Rahwan et al., 2007; Brena et al., 2007; Karunatillake et al., 2006]), among many others. Finally, we have extended **CoalATL** with a goal-based approach as means for forming coalitions. Goals seem important for widely accepted characteristics of intelligent agents like *pro-activeness* and *social ability* [Wooldridge, 2002]. In BDI frameworks, *goals* (or *desires*) and *beliefs* play an important role [Bratman, 1987; Rao and Georgeff, 1991] as well. Finally, it has also been shown that temporal logics like **LTL** and **CTL** can be used as goal specification languages [Bacchus and Kabanza, 1998; Baral and Zhao, 2007; Baral et al., 2001].

### 12.1.4 Resource-Bounded Agents

We pointed out that the modelling and verification of multi-agent systems, in particular the *model checking problem* (i.e. whether a given property holds in a given model), has attracted much attention in recent years. However, resources are usually not taken into account. The only work we are aware of in this direction is [Bulling and Farwer, 2010c,a; Alechina et al., 2009b,a, 2010]. *Resource-bounded coalition logic* (**RBCL**), an extension of coalition logic with resources, is introduced in [Alechina et al., 2009b]. This logic can be seen as a first step towards a multi-agent extension of the resource-bounded tree logics from [Bulling and Farwer, 2010a] under a restricted temporal setting of multiple-step strategies ('sometime in the future'). Only recently, in [Alechina et al., 2010] a multi-agent version (**RBATL**) following the same ideas has been presented. For both logics the authors allow only the consumption of resources which is computationally easier and has a decidable model checking property (cf. Theorem 11.15). The authors of [Alechina et al., 2010] do also propose a sound and complete axiomatisation of their resource-based extension of **ATL** (the logic is called *resource-bounded alternating time temporal logic*).

**RBCL** is used in [Alechina et al., 2009a] to specify and to verify properties about *coalitional resource games* [Wooldridge and Dunne, 2006]. These are games in which agents can cooperate and combine their available resources in order to bring about desired goals.

While most other agent models do not come with an explicit notion of resources, there is some more recent work that takes resources into account. In [Shaw et al., 2008] resources in conjunction with reasoning about an agent's goal-plan tree have been considered. Time, memory, and communication bounds have been studied as resources in [Alechina et al., 2008]. In [Ågotnes and Walther, 2009] the abilities of agents under bounded memory have been analysed. Instead of asking for an arbitrary winning strategy a winning strategy in their setting has to obey given memory limitations.

In order to show decidability of **RTL** we have used ideas of *cover graphs* [Karp and Miller, 1969] and Petri nets. To show the undecidability of the multi-agent variants ideas from automata theory, more precisely counter automata [Hopcroft and Ullman, 1979], have been employed.

## 12.2 Summary and Discussion

### 12.2.1 Summary

In this thesis we have considered formal approaches to model and to reason about rational agents. In Part I we have presented background material which has influenced our work or which has been needed for later sections. The alternating time temporal logics (ATLs) [Alur et al., 2002] have served as the basic logics underlying most of our proposals.

In Section 3.1 we have discussed the close relation between games and logics and have shown how basic extensions of the ATLs can be used to describe game-theoretic solution concepts. We have taken this work as a starting point for our first proposal to reason about rational agents under perfect information. Therefore, we have introduced a logic, *alternating time temporal logic with plausibility* (**ATLP**), generalising several of the existing strategic logics which are related to this game-theoretic context. **ATLP** allows to reason about agents assuming that they act rationally according to a criterion specified as a formula within the object language.

We see the value of this new logic in its expressivity, flexibility, and conceptual simplicity. The logic has been shown to embed several related logics and to be able to express more sophisticated solution concepts due to the possibility to quantify over strategies. Differently to other attempts our notion of rationality has not been hard-coded into the semantics. We have rather allowed to "plug-in" any desirable characterisation one considers rational. Moreover, we have tried to argue that our concept of *plausibility* should be understood in a more general context than just rationality (from the game theoretic point of view). We have also analysed the model checking problems for various fragments of **ATLP**. The general problem has been shown to be **PSPACE**-complete. However, we have also identified interesting fragments that reside between **P** and **PSPACE**. Hence, the difficulty is a tradeoff between the logics expressivity and its complexity.

The logic **ATLP** has been invented to reason about rational agents having *perfect information* about the world. The logic lacks the ability to model agents which are not completely aware of the current state of the world. However, incomplete information and knowledge are very present and important aspects within MASs. To address these issues we have extended **ATLP** accordingly. The resulting logic *constructive strategic logic with plausibility* (**CSLP**) has been shown to be more than the pure fusion of **ATLP** with the incomplete information strategic logic **CSL** from [Jamroga and Ågotnes, 2007]. We have shown that **CSLP** is capable of reasoning about rational agents having incomplete information about the world. Moreover, on top of knowledge and the concept of plausibility / rationality we have defined a neat and non-standard notion of belief.

The interplay between knowledge, time and belief has been motivated by **CTLKP** from [Bulling and Jamroga, 2007a]. **CSLP** can be understood as its multi-agent extension. From a game-theoretical perspective **ATLP** and **CSLP** can be seen as flexible tools to analyse and to reason about extensive form games with perfect and imperfect information, respectively.

In both classes of logics the focal point has been on the power of a predefined group of agents. The aspect of coalition formation has been neglected; still, it is possible to compare the power of predefined coalitions. *Coalitional alternating time temporal logic* (**CoalATL**) is yet another extension of **ATL** with the motivation to reason about the abilities of *rational coalitions*. Logics provide means for modelling and reasoning. Hence, the main conceptual contribution of **CoalATL** has been the merging of **ATL** with an *argumentative approach* to coalition formation.

We have also analysed the model checking complexity modulo the complexity needed for the procedural part given by the argumentation semantics. The way in which time and the dynamics of coalition formation have been combined has resulted in a minimalistic interplay between these concepts. More sophisticated settings taking into account the temporal dynamics of coalition formation provide interesting directions for future research.

The logic **CSLP** captures incomplete information in a specific way namely how agents perceive facts of the world. Different worlds may provide the same information to agents and thus appear indistinguishable. However, **ATL** suggests another interesting possibility to integrate incomplete information. The semantics underlies the “all-or-nothing“ principle. Agents must be successful against *all* opponents' behaviours, including the most destructive ones. Rational agents however may be aware that it often is not rational to assume that the opponents will be able to identify their worst response (from the opponents' point of view). This is the idea of *alternating time temporal logic with probabilistic success* (**pATL**).

Often, agents have some prediction of the opponents behaviour and can act rationally based on a probabilistic notion of success. This allows to model scenarios in which a plan will most probably be successful nevertheless winning cannot be guaranteed. We have examined the logics' relationship to **ATL** and have presented two concrete instantiations in terms of mixed memoryless and behavioural strategies. Finally, the analysis of the model checking complexity has shown that this kind of reasoning does not have to be more difficult than for plain **ATL**, albeit it often is. Although the mathematical theory underlying behavioural strategies is much more sophisticated than for mixed strategies the picture is upside down when it comes to the complexity of model checking. We have proven that the former logic can be verified in **P** and that the problem for **pATL** with mixed strategies is **PP**-hard (thus, in particular **NP** and **coNP**-hard).

Finally, we have considered yet another setting important for rational decisions. In all the considered logics actions did not have any costs assigned to them. However, in practical applications this assumption is often unrealistic. The costs of actions significantly influence the selection of them. Rational agents should take this into account. Moreover, agents may not even have enough resources for some actions in some point in time. The *resource-bounded tree logic* (**RTL**$^*$) and the *resource agent logic* (**RAL**$^*$) addresses these ideas.

The focus has been on the question whether it is possible to verify agents which act rationally in this sense. We have shown that these modification essentially change the picture when it comes to the complexity of model checking. We have identified several variants and have shown that the interplay between specific properties (e.g. perfect recall vs memoryless strategies) may yield an undecidable model checking problem in general.

### 12.2.2 Are These Logics Useful?

The work presented in this thesis is mainly focussed on theoretical aspects of modelling and verifying abilities of rational agents. However, we have also presented model checking results which demonstrate that there is hope to apply specific settings in practise.

In addition to the classical polynomial-time model checking algorithm of **ATL** we have identified a few other settings of our logics with tractable model checking algorithms. For example, we have shown that the verification of rational play can be done in polynomial time over rectangular (Proposition 10.11) or bounded models (Proposition 10.9). Even an extension to probabilistic scenarios resides within **P** (Theorem 10.64).

The results on verifying resource-bounded agents which have been presented here are rather pessimistic. Most cases of the multi-agent logics have been shown undecidable. On the other hand, we were interested in exactly those cases to find out where the boundary of decidability is. Other researchers have shown that there are restricted fragments which are tractable (cf. [Alechina et al., 2010, 2009b]).

We believe that there will be a need to verify properties which go beyond purely temporal specifications (e.g. in security protocols, games, incomplete information scenarios) and which may require concepts (closely) related to settings presented here; although this may not be the case in the near future. It seems also clear that most of the logics proposed in this thesis cannot directly be used due to computational complexity reasons; on the other hand, the few bounded and tractable settings make hope that additional fragments with better computational properties can be identified. But this kind of work is non-trivial and is left for the future.

## 12.3 Outlook

Research on theoretic foundations of MASs provides interesting and manifold opportunities for future work. The more interaction of humans with software programs is required the more an adequate prediction of their behaviours and desires seems to be relevant. The interaction of groups of entities and their ability is crucial for many applications. Logics provide formal tools to model, to design, to reason, and to verify such systems. Hence, we consider them fundamental for the analysis of MASs.

In the following we list possible and more concrete points for future research based on the work presented in this thesis.

- The decidability / undecidability of open fragments of the resource-bounded agent logics is still open.
- The computational complexity of the decidable fragments of the resource-bounded agent logics are still open. It would be interesting to identify tractable fragments.
- A detailed analysis of the model checking complexity and the decidability question for **RTL**$^*$ is still open. It would be particularly interesting to identify constraints that make the logics' model checking problems *efficiently* decidable for a relevant class of MASs.
- From a practical perspective the theoretical model checking results can be used to develop efficient model checkers in order to actually reason about and to verify behaviours of rational agents. Less expressive fragments of the logics presented here can be identified and analysed with respect to their model checking complexity.
- The logic **CoalATL** provides a first attempt to combine strategic reasoning with coalition formation. The dynamic aspect resulting from the temporal dimension of coalition formation has been neglected. We consider this point interesting for future research.
- Other theoretical results for the logics presented here are interesting as well; for example, axiomatisations or the complexity of the satisfiability problem.
- Apart from the resource-bounded cases our analysis has been focussed on extensions of $\mathcal{L}_{ATL}$ with memoryless strategies. An interesting direction for future research is to extend these settings to a richer language and perfect recall.

# Part V

# Appendices

# A

# Miscellaneous

## Contents



## A.1 CTLKP

Logics of knowledge and belief are often too static and inflexible to be applied to real world problems. In particular, they usually offer no concept for expressing that some course of events is *more likely to happen* than another. We address this problem and extend **CTLK** (computation tree logic with knowledge) with a notion of *plausibility*, which allows for practical and counterfactual reasoning. The new logic **CTLKP** (**CTLK** with plausibility) includes also a particular notion of belief. A plausibility update operator is added to this logic in order to change plausibility assumptions dynamically. Furthermore, we examine some important properties of these concepts. In particular, we show that, for a natural class of models, belief is a **KD45** modality. We also show that model checking **CTLKP** is **P**-complete and can be done in time linear with respect to the size of models and formulae.

### A.1.1 Syntax and Semantics

Formally, the language of **CTLKP** is defined as:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{E}\gamma \mid \mathbf{Pl}_a\varphi \mid \mathbf{Ph}\,\varphi \mid K_a\varphi \mid B_a\varphi$$
$$\gamma ::= \bigcirc\varphi \mid \Box\varphi \mid \varphi\mathcal{U}\varphi.$$

For instance, we may claim it is plausible to assume that a shop is closed after the opening hours, though the manager may be physically able to open it at any time: $\mathbf{Pl}_a\mathsf{A}\Box(\mathsf{late} \rightarrow \neg\mathsf{open}) \wedge \mathbf{Ph}\,\mathsf{E}\Diamond(\mathsf{late} \wedge \mathsf{open})$. Another example: It is plausible to expect that an agent will not commit suicide; on the other hand, an agent is (always) physically able to commit that, and it is also plausible to expect that it has this physical ability:

$$\mathbf{Pl}_a\mathsf{A}\Box\neg\mathsf{suicide} \wedge \mathsf{A}\Box\mathbf{Ph}\,\mathsf{E}\Diamond\mathsf{suicide} \wedge \mathbf{Pl}_a\mathsf{A}\Box\mathbf{Ph}\,\mathsf{E}\Diamond\mathsf{suicide}.$$

The semantics of **CTLKP** extends that of **CTLK** as follows. Firstly, we augment the models with *sets of plausible paths*. A *model with plausibility* is given as

$$\mathfrak{M} = \langle Q, R, \sim_1, ..., \sim_k, \Upsilon_1, ..., \Upsilon_k, \pi\rangle,$$

where $\langle Q, R, \sim_1, ..., \sim_k, \pi\rangle$ is a **CTLK** model, and $\Upsilon_a \subseteq \Lambda_{\mathfrak{M}}$ is the set of paths in $\mathfrak{M}$ that are plausible according to agent $a$. If we want to make clear that $\Upsilon_a$ is taken from model $\mathfrak{M}$, we will write $\Upsilon_a^{\mathfrak{M}}$. It seems worth emphasising that this notion of plausibility is *subjective* and *holistic*. It is subjective because $\Upsilon_a$ represents *agent a's subjective view on what is plausible* – and indeed, different agents may have different ideas on plausibility (i.e., $\Upsilon_a$ may differ from $\Upsilon_b$). It is holistic because $\Upsilon_a$ represents agent $a$'s idea of the plausible behavior *of the whole system* (including the behavior of other agents).

*Remark A.1.* In our models, plausibility is also *global*, i.e., plausibility sets do not depend on the state of the system. Investigating systems, in which plausibility is relativised with respect to states (like in [Friedman and Halpern, 1994]), might be an interesting avenue of future work. However, such an approach – while obviously more flexible – allows for potentially counterintuitive system descriptions. For example, it might be the case that path $\lambda$ is plausible in $q = \lambda[0]$, but the set of plausible paths in $q' = \lambda[1]$ is empty. That is, by following plausible path $\lambda$ we are bound to get to an implausible situation. But then, does it make sense to consider $\lambda$ as plausible?

Secondly, we use a non-standard satisfaction relation $\models_P$, which we call *plausible satisfaction*. Let $\mathfrak{M}$ be a **CTLKP** model and $P \subseteq \Lambda_{\mathfrak{M}}$ be an arbitrary subset of paths in $\mathfrak{M}$ (not necessarily any $\Upsilon_a^{\mathfrak{M}}$). $\models_P$ restricts the evaluation of temporal formulae to the paths given in $P$ only. The "absolute" satisfaction relation $\models$ is defined as $\models_{\Lambda_{\mathfrak{M}}}$.

Let $\mathfrak{on}(P)$ be the set of all states that lie on at least one path in $P$, i.e. $\mathfrak{on}(P) = \{q \in Q \mid \exists\lambda \in P \exists i\ (\lambda[i] = q)\}$. Now, the semantics of **CTLKP** can be given through the following clauses:

$\mathfrak{M}, q \models_P p$ iff $q \in \pi(p)$;
$\mathfrak{M}, q \models_P \neg\varphi$ iff $\mathfrak{M}, q \not\models_P \varphi$;
$\mathfrak{M}, q \models_P \varphi \wedge \psi$ iff $\mathfrak{M}, q \models_P \varphi$ and $\mathfrak{M}, q \models_P \psi$;
$\mathfrak{M}, q \models_P \mathsf{E} \bigcirc \varphi$ iff there is a $q$-subpath $\lambda \in P$ such that $\mathfrak{M}, \lambda[1] \models_P \varphi$;
$\mathfrak{M}, q \models_P \mathsf{E}\Box\varphi$ iff there is a $q$-subpath $\lambda \in P$ such that $\mathfrak{M}, \lambda[i] \models_P \varphi$ for every $i \geq 0$;
$\mathfrak{M}, q \models_P \mathsf{E}\varphi\mathcal{U}\psi$ iff there is a $q$-subpath $\lambda \in P$ and $i \geq 0$ such that $\mathfrak{M}, \lambda[i] \models_P \psi$, and $\mathfrak{M}, \lambda[j] \models_P \varphi$ for every $0 \leq j < i$;
$\mathfrak{M}, q \models_P \mathbf{Pl}_a\varphi$ iff $\mathfrak{M}, q \models_{\Upsilon_a} \varphi$;
$\mathfrak{M}, q \models_P \mathbf{Ph}\,\varphi$ iff $\mathfrak{M}, q \models \varphi$;
$\mathfrak{M}, q \models_P K_a\varphi$ iff $\mathfrak{M}, q \models \varphi$ for every $q'$ such that $q \sim_a q'$;
$\mathfrak{M}, q \models_P B_a\varphi$ iff for all $q' \in \mathfrak{on}(\Upsilon_a)$ with $q\mathcal{K}_a q'$, we have that $\mathfrak{M}, q' \models_{\Upsilon_a} \varphi$.

One of the main reasons for using the concept of plausibility is that we want to define agents' *beliefs* out of more primitive concepts–in our case, these are plausibility and indistinguishability–in a way analogous to [Su et al., 2005; Friedman and Halpern, 1994]. If an agent *knows* $\varphi$, it must be "sure" about it. However, *beliefs* of an agent are not necessarily about reliable facts. Still, they should make sense to the agent; if it believes $\varphi$, then the formula should at least hold in all futures that he envisages plausible. Thus, beliefs of an agent may be seen as *things known to it if it disregards all non-plausible possibilities.*

We say that $\varphi$ is $\mathfrak{M}$-*true* ($\mathfrak{M} \models \varphi$) if $\mathfrak{M}, q \models \varphi$ for all $q \in Q_{\mathfrak{M}}$. $\varphi$ is *valid* ($\models \varphi$) if $\mathfrak{M} \models \varphi$ for all models $\mathfrak{M}$. $\varphi$ is $\mathfrak{M}$-*strongly true* ($\mathfrak{M} \models\!\equiv \varphi$) if $\mathfrak{M}, q \models_P \varphi$ for all $q \in Q_{\mathfrak{M}}$ and all $P \subseteq \Lambda_{\mathfrak{M}}$. $\varphi$ is *strongly valid* ($\models\!\equiv \varphi$) if $\mathfrak{M} \models\!\equiv \varphi$ for all models $\mathfrak{M}$.

**Proposition A.2.** *Strong truth and strong validity imply truth and validity, respectively. The reverse does not hold.*

Ultimately, we are going to be interested in normal (not strong) validity, as parameterising the satisfaction relation with a set $P$ is just a technical tool for propagating sets of plausible paths $\Upsilon_a$ into the semantics of nested formulae. The importance of strong validity, however, lies in the fact that $\models\!\equiv \varphi \leftrightarrow \psi$ makes $\varphi$ and $\psi$ completely interchangeable, while the same is not true for normal validity.

**Proposition A.3.** *Let $\Phi[\varphi/\psi]$ denote formula $\Phi$ in which every occurrence of $\psi$ was replaced by $\varphi$. Also, let $\models\!\equiv \varphi \leftrightarrow \psi$. Then for all $M, q, P$: $M, q \models_P \Phi$ iff $M, q \models_P \Phi[\varphi/\psi]$ (in particular, $M, q \models \Phi$ iff $M, q \models \Phi[\varphi/\psi]$).*

*Note that $\models \varphi \leftrightarrow \psi$ does not even imply that $M, q \models \Phi$ iff $M, q \models \Phi[\varphi/\psi]$.*

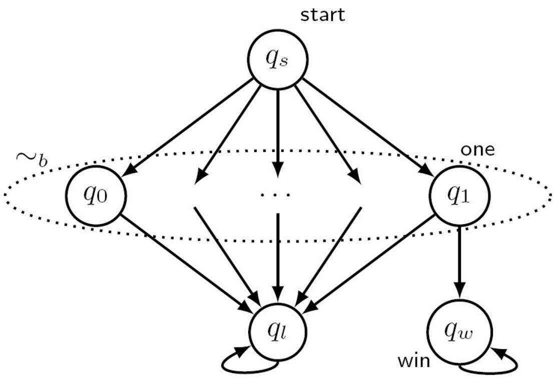



**Fig. A.1.** Guessing Robots game.

*Example A.4 (Guessing robots).* Consider a simple game with two agents $a$ and $b$, shown in Figure A.1. First, $a$ chooses a real number $r \in [0,1]$ (without revealing the number to $b$); then, $b$ chooses a real number $r' \in [0,1]$. The agents win the game (and collect EUR $1,000,000$) if both chose 1, otherwise they lose. Formally, we model the game with a **CTLKP** model $\mathfrak{M}$, in which the set of states $Q$ includes $q_s$ for the initial situation, states $q_r$, $r \in [0,1]$, for the situations after $a$ has chosen number $r$, and "final" states $q_w, q_l$ for the winning and the losing situation, respectively. The transition relation is as follows: $q_s R q_r$ and $q_r R q_l$ for all $r \in [0,1]$; $q_1 R q_w$, $q_w R q_w$, and $q_l R q_l$. Moreover, $\pi(\mathsf{one}) = \{q_1\}$ and $\pi(\mathsf{win}) = \{q_w\}$. Player $a$ has perfect information in the game (i.e., $q \sim_a q'$ iff $q = q'$), but player $b$ does not distinguish between states $q_r$ (i.e., $q_r \sim_b q'_r$ for all $r, r' \in [0,1]$). Obviously, the only sensible thing to do for both agents is to choose 1 (using game-theoretical vocabulary, these strategies are *strongly dominant* for the respective players). Thus, there is only one *plausible* course of events if we assume that our players are rational, and hence $\Upsilon_a = \Upsilon_b = \{q_s q_1 q_w q_w \ldots\}$.

Note that, in principle, the outcome of the game is uncertain: $\mathfrak{M}, q_s \models \neg \mathsf{A}\Diamond\mathsf{win} \wedge \neg\mathsf{A}\Box\neg\mathsf{win}$. However, assuming rationality of the players makes it only plausible that the game must end up with a win: $\mathfrak{M}, q_s \models \mathbf{Pl}_a \mathsf{A}\Diamond\mathsf{win} \wedge \mathbf{Pl}_b \mathsf{A}\Diamond\mathsf{win}$, and the agents believe that this will be the case: $\mathfrak{M}, q_s \models B_a \mathsf{A}\Diamond\mathsf{win} \wedge B_b \mathsf{A}\Diamond\mathsf{win}$. In any of the states $q_r$, agent $b$ believes that $a$ (being rational) has played 1: $\mathfrak{M}, q_r \models B_b \mathsf{one}$ for all $r \in [0,1]$.

### A.1.2 Defining Plausibility with Path Formulae

So far, we have assumed that sets of plausible paths are somehow given within models. In this section we present a dynamic approach where an actual notion of plausibility can be specified in the object language. We want to specify (usually infinite) sets of infinite paths, and we need a finite representation of these structures. A logical solution is given by using path formulae $\gamma$. These formulae describe properties of paths; therefore, a specific formula can be used to characterise a set of paths. For instance, think about a country in Africa where it has never snowed. Then, plausible paths might be defined as ones in which it never snows, i.e., all paths that satisfy $\Box\neg\mathsf{snows}$. Formally, let $\gamma$ be a **CTLK** path formula. We define $|\gamma|_{\mathfrak{M}}$ to be the set of paths that satisfy $\gamma$ in model $\mathfrak{M}$ (when the model is clear from context, the subscript will be omitted):

$$|\bigcirc\varphi|_{\mathfrak{M}} = \{\lambda \mid \mathfrak{M}, \lambda[1] \models \varphi\},$$
$$|\Box\varphi|_{\mathfrak{M}} = \{\lambda \mid \forall i\ (\mathfrak{M}, \lambda[i] \models \varphi)\},$$
$$|\varphi_1 \mathcal{U} \varphi_2|_{\mathfrak{M}} = \{\lambda \mid \exists i\big(\mathfrak{M}, \lambda[i] \models \varphi_2 \wedge \quad \forall j(0 \le j < i \Rightarrow \mathfrak{M}, \lambda[j] \models \varphi_1)\big)\}.$$

Moreover, we define the *plausible paths model update* as follows. Let $\mathfrak{M} = \langle Q, R, \sim_1, ..., \sim_k, \Upsilon_1, ..., \Upsilon_k, \pi\rangle$ be a **CTLKP** model, and let $P \subseteq \Lambda_{\mathfrak{M}}$ be a set of paths. Then $\mathfrak{M}^{a,P} = \langle Q, R, \sim_1, ..., \sim_k, \Upsilon_1, ..., \Upsilon_{a-1}, P, \Upsilon_{a+1}, ..., \Upsilon_k, \pi\rangle$ denotes model $\mathfrak{M}$ with $a$'s set of plausible paths reset to $P$. Note that the set of all paths remains the same in both models because the transition relation does not change, i.e., $\Lambda_{\mathfrak{M}} = \Lambda_{\mathfrak{M}^{a,P}}$.

Now we can extend the language of **CTLKP** with formulae $(\mathbf{set\text{-}pl}_a\ \gamma)\varphi$ with the intuitive reading: Suppose that $\gamma$ exactly characterises the set of plausible paths, then $\varphi$ holds. The formal semantics is given below:

$$\mathfrak{M}, q \models_P (\mathbf{set\text{-}pl}_a\ \gamma)\varphi \text{ iff } \mathfrak{M}^{a,|\gamma|_{\mathfrak{M}}}, q \models_P \varphi.$$

We observe that this update scheme is similar to the one proposed in [Jamroga et al., 2005].

*Remark A.5.* Note, that the set of paths with which the satisfaction relation is annotated does not change after a plausible path update. Consider a **CTLKP** model $\mathfrak{M} = \langle Q, R, \mathcal{K}_1, \ldots, \mathcal{K}_k, \Upsilon_1, \ldots, \Upsilon_k, \pi\rangle$ and statement

$$\mathfrak{M}, q \models_P (\mathbf{set\text{-}pl}_a\ \gamma)\varphi.$$

The semantic rules transform the formula into the equivalent notation

$$\mathfrak{M}^{a,|\gamma|}, q \models_P \varphi.$$

But the set of paths $P$, with which the satisfaction relation is indexed, is still the same as before. If we want set $\Upsilon_a^{\mathfrak{M}^{a,|\gamma|}}$ to be referred to, plausible operator $\mathbf{Pl}_a$ must occur within formula $\varphi$.

### A.1.3 Verification of Plausibility, Time and Beliefs

Clearly, verifying **CTLKP** properties directly against models with plausibility does not make much sense, since these models are inherently infinite; what we need is a finite representation of plausibility sets. Plausibility sets can be defined by path formulae and the update operator ($\mathbf{set\text{-}pl}_a\ \gamma$).

We follow this idea here, studying the complexity of model checking **CTLKP** formulae *against* **CTLK** *models* (which can be seen as a compact representation of **CTLKP** models in which all the paths are assumed plausible), with the underlying idea that plausibility sets, when needed, must be defined explicitly in the object language. Below we sketch an algorithm that model checks **CTLKP** formulae in time linear wrt the size of the model and the length of the formula. This means that we have extended **CTLK** to a more expressive language with no computational price to pay.

First of all, we remove the belief operators by replacing every occurrence of $B_a\varphi$ with $K_a\mathbf{Pl}_a(\mathsf{E}\bigcirc\top\rightarrow\varphi)$. Now, let $\overrightarrow{\gamma}=\langle\gamma_1,...,\gamma_k\rangle$ be a vector of "vanilla" path formulae (one per agent), with the initial vector $\overrightarrow{\gamma_0}=\langle\top,...,\top\rangle$, and $\overrightarrow{\gamma}[\gamma'/a]$ denoting vector $\overrightarrow{\gamma}$, in which $\overrightarrow{\gamma}[a]$ is replaced with $\gamma'$. Additionally, we define $\overrightarrow{\gamma}[0]=\top$. We translate the resulting **CTLKP** formulae to ones without plausibility via function $tr(\varphi)=tr_{\overrightarrow{\gamma_0},0}(\varphi)$, defined as follows:

$$
\begin{aligned}
&tr_{\overrightarrow{\gamma},i}(\mathsf{p})=\mathsf{p},\\
&tr_{\overrightarrow{\gamma},i}(\varphi_1\wedge\varphi_2)=tr_{\overrightarrow{\gamma},i}(\varphi_1)\wedge tr_{\overrightarrow{\gamma},i}(\varphi_2),\\
&tr_{\overrightarrow{\gamma},i}(\neg\varphi)=\neg tr_{\overrightarrow{\gamma},i}(\varphi),\\
&tr_{\overrightarrow{\gamma},i}(K_a\varphi)=K_a\ tr_{\overrightarrow{\gamma},0}(\varphi),\\
&tr_{\overrightarrow{\gamma},i}(\mathbf{Pl}_a\,\varphi)=tr_{\overrightarrow{\gamma},a}(\varphi),\\
&tr_{\overrightarrow{\gamma},i}((\mathbf{set\text{-}pl}_a\ \gamma')\varphi)=tr_{\overrightarrow{\gamma}[\gamma'/a],i}(\varphi),\\
&tr_{\overrightarrow{\gamma},i}(\mathbf{Ph}\,\varphi)=tr_{\overrightarrow{\gamma},0}(\varphi),\\
&tr_{\overrightarrow{\gamma},i}(\bigcirc\varphi)=\bigcirc tr_{\overrightarrow{\gamma},i}(\varphi),\\
&tr_{\overrightarrow{\gamma},i}(\Box\varphi)=\Box tr_{\overrightarrow{\gamma},i}(\varphi),\\
&tr_{\overrightarrow{\gamma},i}(\varphi_1\mathcal{U}\varphi_2)=tr_{\overrightarrow{\gamma},i}(\varphi_1)\mathcal{U}tr_{\overrightarrow{\gamma},i}(\varphi_2),\\
&tr_{\overrightarrow{\gamma},i}(\mathsf{E}\gamma')=\mathsf{E}(\overrightarrow{\gamma}[i]\wedge tr_{\overrightarrow{\gamma},i}(\gamma')).
\end{aligned}
$$

Note that the resulting sentences belong to the logic of $\mathbf{CTLK}^+$, that is $\mathbf{CTL}^+$ (where each path quantifier can be followed by a *Boolean combination* of "vanilla" path formulae) with epistemic modalities. The following proposition justifies the translation.

**Proposition A.6.** *For any* **CTLKP** *formula* $\varphi$ *without* $B_a$*, we have that* $\mathfrak{M},q\models_{\mathbf{CTLKP}}\varphi$ *iff* $\mathfrak{M},q\models_{\mathbf{CTLK}^+}tr(\varphi)$.

In general, model checking $\mathbf{CTL}^+$ (and also $\mathbf{CTLK}^+$) is $\Delta_2^P$-complete. However, in our case, the Boolean combinations of path subformulae are always conjunctions of at most two non-negated elements, which allows us to propose the following model checking algorithm. First, subformulae are evaluated recursively: For each subformula $\psi$ of $\varphi$, the set of states in $\mathfrak{M}$ that satisfy $\psi$ is computed and labeled with a new proposition $\mathsf{p}_\psi$. Now, it is enough to define checking $\mathfrak{M}, q \models \varphi$ for $\varphi$ in which all (state) subformulae are propositions, with the following cases:

Case $\mathfrak{M}, q \models \mathsf{E}(\Box \mathsf{p} \wedge \gamma)$: If $\mathfrak{M}, q \not\models p$, then return **no**. Otherwise, remove from $\mathfrak{M}$ all the states that do not satisfy $p$ (yielding a sparser model $\mathfrak{M}'$), and check the **CTL** formula $\mathsf{E}\gamma$ in $\mathfrak{M}', q$ with any **CTL** model checker.

Case $\mathfrak{M}, q \models \mathsf{E}(\bigcirc \mathsf{p} \wedge \gamma)$: Create $\mathfrak{M}'$ by adding a copy $q'$ of state $q$, in which only the transitions to states satisfying $p$ are kept (i.e., $\mathfrak{M}, q' \models \mathsf{r}$ iff $\mathfrak{M}, q \models \mathsf{r}$; and $q' R_{\mathfrak{M}'} q''$ iff $q R_{\mathfrak{M}} q''$ and $\mathfrak{M}, q'' \models \mathsf{p}$). Then, check $\mathsf{E}\gamma$ in $\mathfrak{M}', q'$.

Case $\mathfrak{M}, q \models \mathsf{E}(\mathsf{p}_1 \mathcal{U} \mathsf{p}_2 \wedge \mathsf{p}_3 \mathcal{U} \mathsf{p}_4)$: Note that this is equivalent to checking $\mathsf{E}(\mathsf{p}_1 \wedge \mathsf{p}_3) \mathcal{U} (\mathsf{p}_2 \wedge \mathsf{E}\mathsf{p}_3 \mathcal{U} \mathsf{p}_4) \vee \mathsf{E}(\mathsf{p}_1 \wedge \mathsf{p}_3) \mathcal{U} (\mathsf{p}_4 \wedge \mathsf{E}\mathsf{p}_1 \mathcal{U} \mathsf{p}_2)$, which is a **CTL** formula.

Other cases: The above cases cover all possible formulas that begin with a path quantifier. For other cases, standard **CTLK** model checking can be used.

**Theorem A.7.** *Model checking* **CTLKP** *against* **CTLK** *models is* **P**-*complete, and can be done in time $O(ml)$, where $m$ is the number of transitions in the model, and $l$ is the length of the formula to be checked. That is, the complexity is no worse than for* **CTLK** *itself.*

## A.2 Discounted **CTL**

In [de Alfaro et al., 2004] *Discounted* CTL (**DCTL**) is proposed. The logic **DCTL** extends **CTL** with a quantitative notion of truth and with the possibility to "discount" the value of truth along a path. The first is addressed by assigning values from $[0,1]$ to proposition instead of Boolean values. The language of **DCTL** is given by

$$\varphi ::= \mathsf{p} \mid \top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \varphi \oplus_c \varphi \mid \mathsf{E}\Diamond_c \varphi \mid \mathsf{E}\Box_c \varphi \mid \mathsf{E}\Delta_c \varphi$$

where $\mathsf{p} \in \Pi$ is a proposition and $c$ (the discount factor) is rational number from $[0,1]$ (its size is measured in binary). It is shown that the universal path quantifier $\mathsf{A}$ can be defined as macro. We will skip details here. The intuition of $\Diamond_c$, $\Box_c$, and $\Delta_c$ is that they returns the supremum, infimum, and average values along a path, respectively. The path quantifiers $\mathsf{E}$ and $\mathsf{A}$ return the supremum and infimum values over all possible paths, respectively.

The authors propose three kinds of models (labeled transition systems, Markov chains and Markov processes) and two semantics (path and fixedpoint semantics). For this thesis we only present the path semantics over *Markov decision processes* (MDPs). For MDPs we refer to Section 4.3.

Let $\mathfrak{M} = (Q, \delta, \Sigma, [\cdot])$ be a Markov decision process. The semantics of formulae is given by $[\![\cdot]\!]$. For a state formulae $\varphi$ we define

$$[\![\varphi]\!] : Q \to [0,1]$$

as follows:

$$\begin{aligned}
[\![\mathsf{p}]\!] &= [\mathsf{p}], \\
[\![\top]\!] &= \mathbf{1}, \\
[\![\bot]\!] &= \mathbf{0}, \\
[\![\neg\varphi]\!] &= \mathbf{1} - [\![\varphi]\!], \\
[\![\varphi_1 \vee \varphi_2]\!] &= \max\{[\![\varphi_1]\!], [\![\varphi_2]\!]\}, \\
[\![\varphi_1 \wedge \varphi_2]\!] &= \min\{[\![\varphi_1]\!], [\![\varphi_2]\!]\}, \\
[\![\varphi_1 \oplus_c \varphi_2]\!] &= (1-c)[\![\varphi_1]\!] + c[\![\varphi_2]\!], \\
[\![\mathsf{E}\gamma]\!](q) &= \sup\{\mathbf{E}_q^{\mathsf{pol}}([\![\gamma]\!]) \mid \mathsf{pol} \in \mathsf{Pol}_{\mathfrak{M}}\}
\end{aligned}$$

and for path formulae $\gamma$ by

$$[\![\gamma]\!] : Q^\omega \to [0,1]$$

where:

$$\begin{aligned}
[\![\Diamond_c\varphi]\!](q_0q_1\dots) &= \sup\{c^i[\![\varphi]\!](q_i) \mid i \geq 0\} \\
[\![\Box_c\varphi]\!](q_0q_1\dots) &= \inf\{1 - c^i(1 - [\![\varphi]\!](q_i)) \mid i \geq 0\} \\
[\![\Delta_c\varphi]\!](q_0q_1\dots) &= \begin{cases} (1-c)\sum_{i\geq 0} c^i[\![\varphi]\!](q_i) & \text{if } c < 1, \\ \lim_{i\geq 0}(\frac{1}{1+1}\sum_{0\leq j\leq i}[\![\varphi]\!](q_j)) & \text{if } c = 1. \end{cases}
\end{aligned}$$

In the definition we have used $\mathbf{1}$ and $\mathbf{0}$ as the functions $Q \ni q \mapsto 1 \in [0,1]$ and $Q \ni q \mapsto 0 \in [0,1]$, respectively.

The expected value $\mathbf{E}_q^{\mathsf{pol}}([\![\gamma]\!])$ of the random variable $[\![\gamma]\!]$ over the $(\mathfrak{M}, \mathsf{pol}, q)$-trajectory space is understood in terms of Definition 4.55 . Finally, we recall the following complexity result.

**Theorem A.8 ([de Alfaro et al., 2004]).** *Given a* **DCTL** *formula* $\varphi$ *and a* MDP $\mathfrak{M}$ *the problem of model checking* $\varphi$ *over* $\mathfrak{M}$ *is in* **P** *with respect to the size of* $\mathfrak{M}$ *and the size of the binary representation of numbers occurring in* $\varphi$. *(The size of the model is given by the number of states and the binary encoding of* $\sum_{\mathsf{p}\in\Pi, q\in Q}[\mathsf{p}](q)$*).*

## A.3 Uniform $\mathbf{ATL_{ir}}$

In this section, we introduce and analyse the logic "uniform $\mathbf{ATL_{ir}}$" ($\mathbf{ATL^{u}_{ir}}$). We use the logic only for technical reasons, namely it provides the intermediate step in the completeness proof of model checking **ATLP**. Still, we believe that the logic is interesting in itself. Moreover, the technique we use for proving completeness is interesting too (and gives insight into the complexity as well as the relationship between the problem we study and known complexity from game theory).

The idea is based on Schobbens' $\mathbf{ATL_{ir}}$ [Schobbens, 2004], i.e., **ATL** for agents with imperfect information and imperfect recall. There, it was assumed that the coalition $A$ in formula $\langle\langle A\rangle\rangle\varphi$ can only use strategies that assign same choices in indistinguishable states (*uniform* strategies). Then, the outcome of every strategy of $A$ was evaluated in every possible behaviour of the remaining agents $\mathbb{A}\mathrm{gt}\setminus A$ (with no additional assumption with respect to that behaviour). In $\mathbf{ATL^{u}_{ir}}$, we assume that the opponents ($\mathbb{A}\mathrm{gt}\setminus A$) are also required to respond *with a uniform memoryless strategy.* The syntax of $\mathbf{ATL^{u}_{ir}}$ is the same as that of **ATL**.

The semantics of $\mathbf{ATL^{u}_{ir}}$ can be defined as follows. Firstly, as models we use *imperfect information concurrent game structures* (ICGSs) from Definition 2.27. Recall that a memoryless strategy $s_A$ is *uniform* if $q \sim_a q'$ implies $s_A|_a(q) = s_A|_a(q')$ for all $q, q' \in Q, a \in A$. To simplify the notation, we define $[q]_a = \{q' \mid q \sim_a q'\}$ to be the class of states indistinguishable from $q$ for $a$; $[q]_A = \bigcup_{a\in A}[q]_a$ collects all the states that are indistinguishable from $q$ for some member of the group $A$; finally, $out(Q', s_A) = \bigcup_{q\in Q'} out(q, s_A)$ collects all the execution paths of strategy $s_A$ from states in set $Q'$.

Now, the semantics is given by the clauses below:

$\mathfrak{M}, q \models_{\mathbf{ATL^{u}_{ir}}} \mathsf{p}$ iff $\mathsf{p} \in \pi(q)$,

$\mathfrak{M}, q \models_{\mathbf{ATL^{u}_{ir}}} \neg\varphi$ iff $\mathfrak{M}, q \not\models_{\mathbf{ATL^{u}_{ir}}} \varphi$,

$\mathfrak{M}, q \models_{\mathbf{ATL^{u}_{ir}}} \varphi \wedge \psi$ iff $\mathfrak{M}, q \models_{\mathbf{ATL^{u}_{ir}}} \varphi$ and $\mathfrak{M}, q \models_{\mathbf{ATL^{u}_{ir}}} \psi$,

$\mathfrak{M}, q \models_{\mathbf{ATL^{u}_{ir}}} \langle\langle A\rangle\rangle \bigcirc \varphi$ iff there is a uniform strategy $s_A$ such that, for each uniform counterstrategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$, and $\lambda \in out([q]_A, \langle s_A, t_{\mathbb{A}\mathrm{gt}\setminus A}\rangle)$,[1] we have $\mathfrak{M}, \lambda[1] \models_{\mathbf{ATL^{u}_{ir}}} \varphi$,

$\mathfrak{M}, q \models_{\mathbf{ATL^{u}_{ir}}} \langle\langle A\rangle\rangle \Box\varphi$ iff there is a uniform strategy $s_A$ such that, for each uniform counterstrategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$, and $\lambda \in out([q]_A, \langle s_A, t_{\mathbb{A}\mathrm{gt}\setminus A}\rangle)$, we have $\mathfrak{M}, \lambda[i] \models_{\mathbf{ATL^{u}_{ir}}} \varphi$ for all $i = 0, 1, ...$,

$\mathfrak{M}, q \models_{\mathbf{ATL^{u}_{ir}}} \langle\langle A\rangle\rangle \varphi\mathcal{U}\psi$ iff there is a uniform strategy $s_A$ such that, for each uniform counterstrategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$, and $\lambda \in out([q]_A, \langle s_A, t_{\mathbb{A}\mathrm{gt}\setminus A}\rangle)$, there is $i \in \mathbb{N}_0$ with $\mathfrak{M}, \lambda[i] \models_{\mathbf{ATL^{u}_{ir}}} \psi$, and $\mathfrak{M}, \lambda[j] \models_{\mathbf{ATL^{u}_{ir}}} \varphi$ for all $0 \le j < i$.

---

[1] Note that the definition of concurrent game structures, that we use after [Alur et al., 2002], implies that CGS are deterministic, so there is in fact exactly one such path $\lambda$.

As before the logic $\mathbf{ATL}^u_{\mathbf{ir}}$ is formally defined by $(\mathcal{L}_{ATL}, \models_{\mathbf{ATL}^u_{\mathbf{ir}}})$.

## A.4 Bargaining with Discount

In Example 3.12 we presented bargaining with discount. After each round the worth of the goods is reduced by $\delta_i$. In round $t$ the goods have a value of $r(\delta_i^t)$. Because we use a rounding function $r$, there is a minimal round $T$ such that $r(\delta_i^{T+1}) = 0$ for $i = 1$ or $i = 2$. We can treat this case as finite horizon bargaining game [Ståhl, 1972; Mas-Colell et al., 1995].

Now, consider the case that $a_i$'s opponent, denoted by $a_{-i}$, is the offerer in $T$. It can offer 0 and $a_i$ should accept, because in the next round the goods are worthless for $a_i$.

On the other hand, if $a_i$ is offerer in $T$ we have to distinguish two cases. If $r(\delta_{-i}^{T+1}) = 0$ then following the same reasoning as before $a_i$ can offer 0 to $a_{-i}$. In the other case, namely $r(\delta_{-i}^{T+1}) \neq 0$, we consider the subsequent round $T+1$ in which $a_{-i}$ takes the role as offerer and can successfully offer 0 to $i$.

Now, it is possible to solve the game starting from the end. Solutions for $\delta_1 = \delta_2$ can be found in the literature [Mas-Colell et al., 1995]. Here, we recall the idea for different discount rates.

At first, let $a_1$ be the last offerer and $r(\delta_2^{T+1}) = 0$. This implies, that $T$ is even (the initial round is 0). In $T$, $a_1$ offers $\langle 1, 0\rangle$ and $a_2$ accepts. Knowing this, in $T-1$ agent $a_2$ can offer $\langle \delta_1, 1-\delta_1\rangle$, since in the next round the value of the good for $a_1$ would become reduced by $\delta_1$. Following the same reasoning, in $T-2$ $a_1$ could successfully offer $\langle 1-\delta_2(1-\delta_1), \delta_2(1-\delta_1)\rangle$. Finally, in round $t=0$ $a_1$ can offer $\langle \zeta, 1-\zeta\rangle$ where

$$\zeta := (1-\delta_2)\sum_{i=0}^{\frac{T}{2}-1}(\delta_1\delta_2)^i + (\delta_1\delta_2)^{\frac{T}{2}} = (1-\delta_2)\frac{1-(\delta_1\delta_2)^{\frac{T}{2}}}{1-\delta_1\delta_2} + (\delta_1\delta_2)^{\frac{T}{2}}.$$

Secondly, consider the case in which $a_2$ is the last offerer in $T$ and $r(\delta_1^{T+1}) = 0$. This time $T$ is odd but the reasoning stays the same. In round 0 $a_1$ can offer $\langle \zeta', 1-\zeta'\rangle$ where

$$\zeta' := (1-\delta_2)\frac{1-(\delta_1\delta_2)^{\frac{T+1}{2}}}{1-\delta_1\delta_2}.$$

# B

# Proofs

## Contents



In this section we give some detailed proofs which are too lengthy or technical for putting them into the main matter.

## B.1 Rational Play: ATLP

**Proposition 6.32 ($\rightsquigarrow$ page 113). ATLP** *embeds* **ATLI**.

*Proof.* For an **ATLI**-model $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, o, \mathcal{I}, \mathfrak{Str}, [\cdot] \rangle$, we construct the corresponding concurrent game structure with plausibility $TR(\mathfrak{M}) = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act,\ d, o, \Upsilon, \Omega, [\![\cdot]\!] \rangle$ with the set of plausible strategy profiles $\Upsilon = \{s \in \Sigma \mid s \text{ is consistent with } \mathcal{I}\}$, plausibility terms $\Omega = \{\omega_\sigma \mid \sigma \in \mathfrak{Str}\} \cup \{\omega_\top\}$, and their denotation $[\![\omega_\top]\!]^q = \Sigma$ and $[\![\omega_\sigma]\!]^q = \{s \in \Sigma \mid s|_a = [\sigma]\}$ for each $\sigma \in \mathfrak{Str}_a$ and $a \in \mathbb{A}\mathrm{gt}$.

For an $\mathcal{L}_{ATLI}$-formula $\varphi$, we construct its $\mathcal{L}_{ATLP}$ translation by transforming strategic assumptions (about agents' intentions) imposed by $(\mathbf{str}_a\sigma)$ to plausibility assumptions (about strategy profiles that can be plausibly played) defined by $(\mathbf{set\text{-}pl}\ \omega_\sigma)$ and applying them to the appropriate set of agents (i.e., those for whom strategic assumptions have been defined). Formally, the translation is defined as $tr(\varphi) = \mathbf{Pl}\ tr_{\langle\omega_\top,\dots,\omega_\top\rangle}(\varphi)$, where $tr_{\langle\omega_1,\dots,\omega_k\rangle}$ is defined as follows:

$$
\begin{aligned}
tr_{\langle\omega_1,\dots,\omega_k\rangle}(\mathsf{p}) &= \mathsf{p},\\
tr_{\langle\omega_1,\dots,\omega_k\rangle}(\neg\varphi) &= \neg tr_{\langle\omega_1,\dots,\omega_k\rangle}(\varphi),\\
tr_{\langle\omega_1,\dots,\omega_k\rangle}(\varphi_1\wedge\varphi_2) &= tr_{\langle\omega_1,\dots,\omega_k\rangle}(\varphi_1)\wedge tr_{\langle\omega_1,\dots,\omega_k\rangle}(\varphi_2),\\
tr_{\langle\omega_1,\dots,\omega_k\rangle}(\langle\!\langle A\rangle\!\rangle \bigcirc \varphi) &= \langle\!\langle A\rangle\!\rangle \bigcirc tr_{\langle\omega_1,\dots,\omega_k\rangle}(\varphi),\\
&\quad\text{(for } \langle\!\langle A\rangle\!\rangle\Box\varphi \text{ and } \langle\!\langle A\rangle\!\rangle\varphi_1\mathcal{U}\varphi_2 \text{ analogously)}\\
tr_{\langle\omega_1,\dots,\omega_k\rangle}((\mathbf{str}_a\sigma'_a)\varphi) &= (\mathbf{set\text{-}pl}\ \overrightarrow{\omega})tr_{\overrightarrow{\omega}}(\varphi),\\
\text{where}\quad \overrightarrow{\omega} &= \langle\omega_1,\dots,\omega_{\sigma'_a},\dots,\omega_k\rangle.
\end{aligned}
$$

For **ATLI**, $\langle\!\langle A\rangle\!\rangle\gamma$ holds iff $\gamma$ can be enforced *against each response strategy from* $\mathbb{A}\mathrm{gt}\setminus A$. Thus, e.g., $\mathfrak{M},q\models_{\mathsf{ATLI}}(\mathbf{str}_a\sigma_a)\langle\!\langle A\rangle\!\rangle\Box\mathsf{p}$ iff $TR(\mathfrak{M}),q\models_{\mathsf{ATLP}}$ $\mathbf{Pl}\,(\mathbf{set\text{-}pl}\ \langle\omega_\top,\dots,\omega_{\sigma_a},\dots,\omega_\top\rangle)\langle\!\langle A\rangle\!\rangle\Box\mathsf{p}$.

The proof is done by structural induction on $\varphi$. The basic cases for propositions, conjunction, and negation are straightforward. Let $\mathfrak{M}$ be an **ATLI** model and $\mathfrak{N} := TR(\mathfrak{M})$ the constructed CGSP where the plausibility term $\langle\omega_1,\dots,\omega_k\rangle$ describes the intention relation from $\mathfrak{M}$ in the obvious way.

$\varphi\equiv\langle\!\langle A\rangle\!\rangle\bigcirc\psi$. $\mathfrak{N},q\models_{\mathbb{A}\mathrm{gt}} tr_{\langle\omega_1,\dots,\omega_k\rangle}(\langle\!\langle A\rangle\!\rangle\bigcirc\psi)$ iff $\mathfrak{N},q\models_{\mathbb{A}\mathrm{gt}}\langle\!\langle A\rangle\!\rangle\bigcirc tr_{\langle\omega_1,\dots,\omega_k\rangle}(\psi)$ iff there is a $\mathbb{A}\mathrm{gt}$-plausible strategy $s_A$ such that for all $\lambda\in out(q,s_A,\mathbb{A}\mathrm{gt})$ we have that $\mathfrak{N},\lambda[1]\models_{\mathbb{A}\mathrm{gt}} tr_{\langle\omega_1,\dots,\omega_k\rangle}(\psi)$ iff there is a strategy $s_A$ consistent with $\mathcal{I}$ such that for all $\lambda\in out(q,s_A,\mathbb{A}\mathrm{gt})$ we have that $\mathfrak{N},\lambda[1]\models_{\mathbb{A}\mathrm{gt}} tr_{\langle\omega_1,\dots,\omega_k\rangle}(\psi)$ iff there is a strategy $s_A$ consistent with $\mathcal{I}$ such that for all $\lambda\in out^{\mathcal{I}}(q,s_A)$ we have that $\mathfrak{M},\lambda[1]\models_{\mathbb{A}\mathrm{gt}}\psi$ iff $\mathfrak{M},q\models_{\mathsf{ATLI}}\langle\!\langle A\rangle\!\rangle\bigcirc\psi$.

$\varphi\equiv\langle\!\langle A\rangle\!\rangle\Box\psi$ and $\langle\!\langle A\rangle\!\rangle\psi\mathcal{U}\chi$. Analogously.

$\varphi\equiv(\mathbf{str}_a\sigma'_a)\psi$. $\mathfrak{N},q\models_{\mathbb{A}\mathrm{gt}} tr_{\langle\omega_1,\dots,\omega_k\rangle}((\mathbf{str}_a\sigma'_a)\psi)$ iff $\mathfrak{N},q\models_{\mathbb{A}\mathrm{gt}}(\mathbf{set\text{-}pl}\ \overrightarrow{\omega})tr_{\overrightarrow{\omega}}(\psi)$ where $\overrightarrow{\omega}=\langle\omega_1,\dots,\omega_{\sigma'_a},\dots,\omega_k\rangle$ iff $\mathfrak{N}^{\overrightarrow{\omega}},q\models_{\mathbb{A}\mathrm{gt}} tr_{\overrightarrow{\omega}}(\psi)$ where $\mathfrak{N}^{\overrightarrow{\omega}}$ is the new model that equals $\mathfrak{N}$ but the set of plausible strategies is set to $\overrightarrow{\omega}=\langle\omega_1,\dots,\omega_{\sigma'_a},\dots,\omega_k\rangle$ iff $\mathfrak{M}',q\models_{\mathsf{ATLI}}\psi$ where the intention relation in $\mathfrak{M}'$ is described by $\overrightarrow{\omega}=\langle\omega_1,\dots,\omega_{\sigma'_a},\dots,\omega_k\rangle$ iff $\mathfrak{M},q\models_{\mathsf{ATLI}}(\mathbf{str}_a\sigma'_a)\psi$ where the intention relation in $\mathfrak{M}$ is described by $\overrightarrow{\omega}=\langle\omega_1,\dots,\omega_k\rangle$.

■

**Proposition 6.36 (⇝ page 115).** **GLP** *can be embedded in* **ATLP**.

*Proof.* For the translation of models, we transform game trees of **GLP** to concurrent game structures using the construction from Section 3.2.2 (particularly cf. Def. 3.11), and transform the CGSs to CGSPs by taking $\Upsilon = \Sigma$ and $\Omega = \emptyset$. Then, we use the following translation of $\mathcal{L}_{GLP}$-formulae:

$$\begin{aligned}
tr(\varphi) &= \mathbf{Pl}\; tr_{(\mathbf{set\text{-}pl}\;\sigma.\top)}(\varphi),\\
tr_\omega(\mathsf{p}) &= \mathsf{p}, \quad tr_\omega(\neg\varphi) = \neg tr_\omega(\varphi), \quad tr_\omega(\varphi \vee \psi) = tr_\omega(\varphi) \vee tr_\omega(\psi),\\
tr_\omega(\Box\varphi_0) &= \langle\langle\emptyset\rangle\rangle \Diamond\varphi_0,\\
tr_\omega([A:\varphi_0]\psi) &= (\mathbf{set\text{-}pl}\;\omega_{A,\varphi_0}) tr_{\omega_{A,\varphi_0}}(\psi),\\
\text{where}\quad \omega_{A,\varphi_0} &= \sigma.\mathbf{Pl}\,(\mathbf{set\text{-}pl}\;\omega)\langle\langle\emptyset\rangle\rangle\Box\big(\mathit{plausible}(\sigma) \wedge \mathit{prefers}(A,\sigma,\varphi_0)\big),\\
\mathit{plausible}(\sigma) &\equiv (\mathbf{refn\text{-}pl}\;\sigma)\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle \bigcirc \top,\\
\mathit{prefers}(A,\sigma,\varphi_0) &\equiv \langle\langle A\rangle\rangle\Diamond\varphi_0 \rightarrow (\mathbf{refn\text{-}pl}\;\sigma[A])\langle\langle\emptyset\rangle\rangle\Diamond\varphi_0.
\end{aligned}$$

That is, with each subsequent preference operator $[A : \varphi_0]$, only those from the (currently) plausible strategy profiles are selected that are preferred by $A$. The preference is based on the (subgame perfect) enforceability of the outcome $\varphi_0$ at the end of the game: If $\varphi_0$ can be enforced at all, then $A$ prefers strategies that do enforce it.

**Lemma B.1.** *Let $\Gamma$ be a* **GLP***-model, $\mathfrak{M} = TR(\Gamma)$ its corresponding* CGSP *and let $\mathfrak{M}'$ be equal to $\mathfrak{M}$ but the set of plausible strategies given by $[\![\omega_{A,\varphi_0}]\!]$. Moreover, let $\mathfrak{M}''$ be the restriction of $\mathfrak{M}'$ obtained if each state not reachable from $\emptyset$ by any plausible strategy from $[\![\omega_{A,\varphi_0}]\!]_{\mathfrak{M}'}$ is removed and in which $\Upsilon_{\mathfrak{M}''} = \Sigma$ and $\Omega_{\mathfrak{M}''} = \emptyset$. Then, we have that*

$$TR(Up(\Gamma, A, \varphi_0)) = \mathfrak{M}''.$$

*Proof.* Suppose there is no subgame perfect strategy of $A$ enforcing $\varphi_0$. Then, $Up(\Gamma, A, \varphi_0) = \Gamma$ and $[\![\omega_{A,\varphi_0}]\!] = \Sigma$. Each state in $\mathfrak{M}'$ is reachable by some strategy from $\Sigma$ and hence $TR(Up(\Gamma, A, \varphi_0)) = \mathfrak{M}''$.

We now consider the case where there is a most general subgame perfect strategy. Suppose $\mathfrak{N} := TR(Up(\Gamma, A, \varphi_0)) \neq \mathfrak{M}''$. Let $h$ be a state reachable in $\mathfrak{N}$ from $\emptyset$ that is not reachable in $\mathfrak{M}''$. Then, $h$ is reachable in $\Gamma$ from $\emptyset$ by the most general subgame perfect strategy and there is no strategy $s_A \in [\![\omega_{A,\varphi_0}]\!]$ such that $h$ is reachable in $\mathfrak{M}''$. Let $s'_A$ be some *Ir*-strategy obtained from $s_A$ by chosing specific choices such that $h$ is still reachable if $s'_A$ is executed in $\emptyset$. (Note that the most general subgame perfect strategy can be seen as a set of *ir*-strategies). Then, we have that $s'_A \in [\![\omega_{A,\varphi_0}]\!]$. Contradiction.

For the other direction, suppose $h$ is reachable from $\emptyset$ in $\mathfrak{M}''$ but not in $\mathfrak{N}$. Then, let $s_A \in [\![\omega_{A,\varphi_0}]\!]$ be some strategy such that $h$ is reachable. As $s_A$ guarantees $\Diamond\varphi_0$ all the states reachable by $s_A$ are also reachable by the most general subgame perfect strategy; otherwise, it would contradict the most general character of such a strategy. ■

Now, we have that $\Gamma \models_{\mathsf{GLP}} \varphi$ iff $TR(\Gamma), \emptyset \models_{\mathsf{ATLP}} tr(\varphi)$.[1] The proof is done by structural induction. The cases for the Boolean connectives are done as usual and are omitted here.

Case $\Box\varphi_0$: $\Gamma \models \Box\varphi_0$ iff $\forall h \in Z(H) : \pi, h \models \varphi_0$ iff all reachable leaf nodes of $TR(\Gamma)$ satisfy $\varphi_0$ iff $TR(\Gamma), \emptyset \models_{\mathsf{ATLP}} \langle\langle\emptyset\rangle\rangle\Diamond\varphi_0$.

Case $[A : \varphi_0]\psi$: $\Gamma \models [A : \varphi_0]\psi$ iff $Up(\Gamma, A, \varphi_0) \models \psi$ iff $TR(Up(\Gamma, A, \varphi_0)), \emptyset \models_{\mathbb{A}\mathrm{gt}} tr_{\omega_{A,\varphi_0}}(\psi)$ (Lemma B.1) iff $TR(\Gamma), \emptyset \models_{\mathbb{A}\mathrm{gt}}$ (**set-pl** $\omega_{A,\varphi_0})tr_{\omega_{A,\varphi_0}}(\psi)$.

■

## B.2 ATL with Probabilistic Success

**Proposition 7.52 (⇝ page 168).** *Function $holds_\gamma$ is $\mathcal{S}(q)$-measurable and $\mu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}$-integrable for $\mathcal{L}_{pATL}$-path formulae $\gamma$.*

*Proof.* In particular, we have to show that $holds_\gamma^{-1}(A) := \{\lambda \in \Lambda(q) \mid holds_\gamma(\lambda) \in A\}$ is measurable for each $A \subseteq \{0, 1\}$ (i.e. $holds_\gamma^{-1}(A) \in \mathcal{S}(q)$). The cases $\emptyset$ and $\{0, 1\}$ are trivial. The case for $\{0\}$ is clear if we have shown it for $A = \{1\}$ (cf. property (ii) of $\sigma$-algebras, Section 4.3). Therefore, let $f_\gamma := holds_\gamma^{-1}(\{1\})$. The proof proceeds by structural induction on $\gamma$.
**I. Case "□": (i)** Let $\gamma = \Box p$ where $p$ is a propositional logic formula (e.g. $p = \mathsf{r} \wedge \neg\mathsf{s}$). We define $L_n^{\Box p} := \{\lambda \in \Lambda(q) \mid \forall i \in \mathbb{N}_0(0 \leq i < n \rightarrow \mathfrak{M}, \lambda[i] \models p\}$. We have that each $L_n^{\Box p} \in \mathcal{F}^n(q) \subseteq \mathcal{S}(q)$ and that $\bigcap_{n\in\mathbb{N}} L_n^{\Box p} = f_\gamma$; hence, also that $f_\gamma \in \mathcal{S}(q)$ because of property (ii) and (iv) of $\sigma$-algebras (cf. Section 4.3). That $f_\gamma$ is integrable follows from Lebesgue's Dominated Convergence Theorem: $f_\gamma$ is measurable and $|f_\gamma|$ is bounded by the $\mu^{s_A}_{\beta_{\mathbb{A}\mathrm{gt}\setminus A}}$-integrable (constant) function 1. **(ii)** Let $\gamma = \Box\langle\langle B\rangle\rangle^p_{\omega'}\gamma'$ and suppose $f_{\gamma'}$ is already proven to be integrable. Then, $L_n^{\Box\langle\langle B\rangle\rangle^p_{\omega'}\gamma'}$ can be defined in the same way as above. **(iii)** Suppose that for each sub path formula $\gamma'$ contained in $\varphi_1$ and $\varphi_2$ we have proven that $f_{\gamma'}$ is integrable, then $L_n^\gamma$ can be defined in the same way as above for $\gamma = \Box\neg\varphi_1$ and $\gamma = \Box(\varphi_1 \wedge \varphi_2)$.
**II. Case "◯":** Similar to I(i) we define $L_n^{\bigcirc p} := \{\lambda \in \Lambda(q) \mid \mathfrak{M}, \lambda[1] \models p\}$. Then, we have that $\bigcup_{n\in\mathbb{N}} L_n^{\bigcirc p} = f_{\bigcirc p} \in \mathcal{S}(q)$. The rest of the proof is done analogously to I.
**III. Case "$\mathcal{U}$":** Here, we also just consider the part corresponding to I(i). We set $L_n^{p\mathcal{U}q} := \{\lambda \in \Lambda(q) \mid \exists j(0 \leq j < n \rightarrow (\mathfrak{M}, \lambda[j] \models q \wedge \forall i \in \mathbb{N}_0(0 \leq i < j \rightarrow \mathfrak{M}, \lambda[i] \models p))\}$; then, we have that $\bigcup_{n\in\mathbb{N}_0} L_n^{p\mathcal{U}q} = f_{p\mathcal{U}q} \in \mathcal{S}(q)$. ■

---

[1] Again, $\emptyset$ denotes the position with empty history, i.e., the initial state of the game.

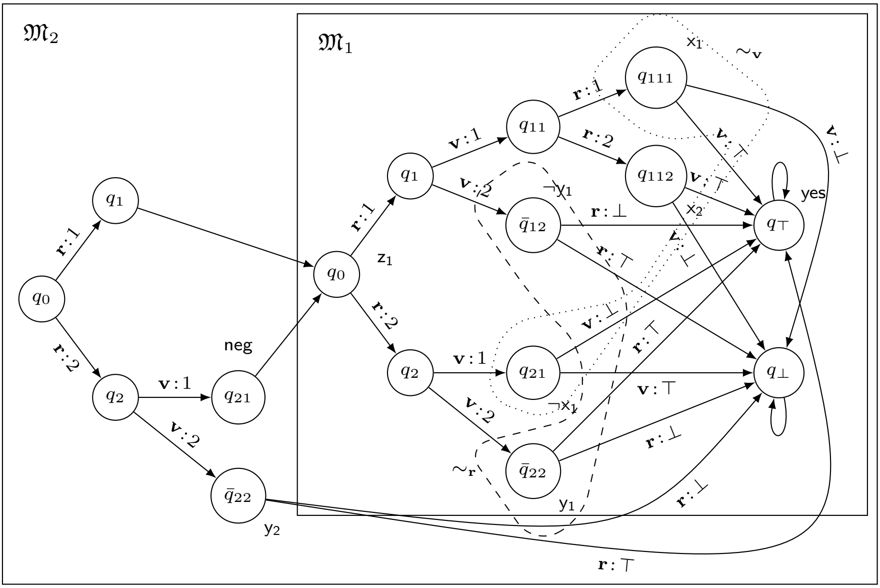



**Fig. B.1.** CEGS $\mathfrak{M}_2$ for $\varphi_1 \equiv ((x_1 \wedge x_2) \vee \neg y_1) \wedge (\neg x_1 \vee y_1)$, $\varphi_2 \equiv z_1 \wedge (\neg z_1 \vee y_2)$.

## B.3 Model Checking Uniform ATL$_{\text{ir}}$

We show the lower bound by a reduction of SNSAT$_2$, a typical $\mathbf{\Delta_3^P}$-complete problem (cf. Definition 4.15). In this section we focus on SNSAT$_2$ where we set $X_r^1 = X_r = \{x_{1,r}, ..., x_{k,r}\}$ and $X_r^2 = Y_r = \{y_{1,r}, ..., y_{k,r}\}$.

Our reduction of SNSAT$_2$ is an extension of the reduction of SNSAT presented in [Jamroga and Dix, 2006, 2008]. That is, we construct the ICGS $\mathfrak{M}_r$ corresponding to $z_r$ with two players: *verifier* **v** and *refuter* **r**. The ICGS is turn-based, that is, every state is "governed" by a single player who determines the next transition. Each subformula $\chi_{i_1 \ldots i_l}$ of $\varphi_r$ has a corresponding state $q_{i_1 \ldots i_l}$ in $\mathfrak{M}_r$. If the outermost logical connective of $\varphi_r$ is $\wedge$, the refuter decides at $q_0$ which subformula $\chi_i$ of $\varphi_r$ is to be satisfied, by proceeding to the "subformula" state $q_i$ corresponding to $\chi_i$. If the outermost connective is $\vee$, the verifier decides which subformula $\chi_i$ of $\varphi_r$ will be attempted at $q_0$. This procedure is repeated until all subformulae are single literals. The states corresponding to literals are called "proposition" states.

The difference from the construction from [Jamroga and Dix, 2006, 2008] is that formulae are in positive normal form (rather than CNF) and that we have two kinds of "proposition" states now: $q_{i_1 \ldots i_l}$ refers to a literal consisting of some $x \in X_r$ and is governed by **v**; $\bar{q}_{i_1 \ldots i_l}$ refers to some $y \in Y_r$ and will be governed by **r**. Now, the values of the underlying propositional variables $x, y$

are declared at the "proposition" states, and the outcome is computed. That is, if $\mathbf{v}$ executes $\top$ for a positive literal, i.e. $\chi_{i_1\ldots i_l} = x$, (or $\bot$ for $\chi_{i_1\ldots i_l} = \neg x$) at $q_{i_1\ldots i_l}$, then the system proceeds to the "winning" state $q_\top$; otherwise, the system goes to the "sink" state $q_\bot$. For states $\bar{q}_{i_1\ldots i_l}$ the procedure is analogous. Models corresponding to subsequent $z_r$ are nested like in Figure B.1.[2] "Proposition" states referring to the same variable $x$ are indistinguishable for $\mathbf{v}$ (so that he has to declare the same value of $x$ in all of them), and the states referring to the same $y$ are indistinguishable for $\mathbf{r}$. A sole $\mathsf{ATL}_{\mathsf{ir}}^{u}$ proposition yes holds only in the "winning" state $q_\top$. As in [Jamroga and Dix, 2006, 2008], we have the following result which concludes the reduction.

**Proposition B.2.** *The above construction shows a polynomial reduction of* SNSAT$_2$ *to model checking* $\mathsf{ATL}_{\mathsf{ir}}^{u}$ *in the following sense. Let*

$$\Phi_1 \equiv \langle\langle \mathbf{v} \rangle\rangle (\neg \mathsf{neg}) \mathcal{U} \mathsf{yes}, \quad \textit{and}$$
$$\Phi_r \equiv \langle\langle \mathbf{v} \rangle\rangle (\neg \mathsf{neg}) \mathcal{U} (\mathsf{yes} \vee (\mathsf{neg} \wedge \langle\langle \varnothing \rangle\rangle \bigcirc \neg \Phi_{r-1})) \quad \textit{for } r = 2, \ldots, p.$$

*Then, we have* $z_p$ *iff* $\mathfrak{M}_p, q_0^p \models_{\mathsf{ATL}_{\mathsf{ir}}^{u}} \Phi_p$.

As for the upper bound, we note that there is a straightforward $\mathbf{\Delta_3^P}$ algorithm that model checks formulae of $\mathsf{ATL}_{\mathsf{ir}}^{u}$: when checking $\langle\langle A \rangle\rangle T\varphi$ in $\mathfrak{M}, q$, it first recursively checks $\varphi$ (bottom-up), and labels the states where $\varphi$ held with a special proposition yes. Then, the algorithm guesses a uniform strategy $s_A$ and calls an oracle that guesses a uniform counterstrategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$. Finally, it trims $\mathfrak{M}$ according to $\langle s_A, t_{\mathbb{A}\mathrm{gt}\setminus A}\rangle$, and calls a **CTL** model checker to check formula $\mathsf{A}T\mathsf{yes}$ in state $q$ of the resulting model. This gives us the following result.

**Theorem 10.15 (⇝ page 229).** *Model checking* $\mathsf{ATL}_{\mathsf{ir}}^{u}$ *is* $\mathbf{\Delta_3^P}$*-complete with respect to the number of transitions in the model and the length of the formula. It is* $\mathbf{\Delta_3^P}$*-complete even for turn based* ICGS *with at most two agents.*

## B.4 Model Checking ATLP

### B.4.1 Results in Section 10.1.1

**Proposition 10.4 (⇝ page 226).** *Let* $\mathfrak{M}$ *be a well-behaved* CGSP*,* $q$ *a state in* $\mathfrak{M}$*, and* $\varphi$ *a formula of* $\mathcal{L}_{ATLP}^{base}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$*. Then* $\mathfrak{M}, q \models \varphi$ *iff* $mcheckATLP(\mathfrak{M}, q, \varphi)$*. The algorithm runs in time* $\mathbf{\Delta_3^P}$ *with respect to the number of transitions in the model and the length of the formula.*

[2] All states in the model for $z_r$ are additionally indexed by $r$.

*Proof.* Function *mcheck* is called recursively, at most $l$ times. All cases apart from $\varphi \equiv \langle\langle A \rangle\rangle \bigcirc \psi$ where $\psi$ includes no $\langle\langle C \rangle\rangle$ (analogously for the other temporal operators) can be performed in polynomial time. Now, there is a nondeterministic Turing machine $A_B$ which implements function *beatable*: Firstly, it guesses a strategy $t$ possibly together with another witness necessary for *plausiblestrat* (by assumption the latter is in **NP**) and verifies if $t$ is plausible, the verification can be done in polynomial time (by the same assumption). Finally, if $t$ is plausible $A_B$ has to perform **CTL** model checking which lies in **P**.

It remains to show that there is a nondeterministic oracle Turing machine $A_S$ with oracle $A_B$ implementing *solve*. (Formally, the machine requires two oracles, one answering the question whether $s$ is plausible, and the other is given by $A_B$. However, the former is computationally less expensive than the latter and can be ignored since we are interested in the oracle with the highest complexity.) $A_S$ works as follows: Firstly, it guesses a profile $s$ (again possibly together with a witness for *plausiblestrat*); secondly, it verifies whether $s$ is plausible and then calls oracle $A_B$ and inverts its answer. Altogether, there are polynomial many calls to machine $A_S^{A_B} \in \mathbf{NP}^{\mathbf{NP}}$. This renders the algorithm to be in $\mathbf{\Delta_3^P}$. ■

**Proposition 10.9 (⇝ page 227).** *Let $c \in \mathbb{N}$ be a constant. Model checking $\mathcal{L}_{ATLP}^{base}$ formulae with respect to the class of well-behaved bounded models $\mathfrak{M}^c$ can be done in polynomial time with respect to the number of transitions in the model and the length of the formula.*

*Proof.* We modify the original **ATL** model checking procedure as follows. Consider the formula $\varphi \equiv \langle\langle A \rangle\rangle \gamma$ where $\gamma$ is a pure **ATL** path formula. Let $B$ be the set of agents assumed to play plausibly and let $\Upsilon \neq \Sigma$ be the current set of plausible strategies described by some term and state. For each $s_B \in \Upsilon|_B$ we remove from $\mathfrak{M}$ all transitions which cannot occur according to $s_B$, yielding model $\mathfrak{M}^{s_B}$, and check whether $\mathfrak{M}^{s_B}, q \models_{\mathsf{ATL}} \langle\langle A \rangle\rangle \gamma$. We proceed like this for all $s \in \Upsilon|_B$ (there are only polynomially many). This procedure is incorporated into our **ATLP** model checking algorithm and applied bottom up. ■

### B.4.2 Results in Section 10.1.4

#### Upper Bounds

First, we recall a basic complexity result that will be used in the rest of this section. Then, we present proofs of upper bounds for model checking $\mathcal{L}_{ATLP}^{k}$ for pure CGSs and well-behaved CGSPs.

*Remark B.3.* A relation $R \subseteq \times_{i=1}^{k+1} \Sigma^*$ ($k \geq 1$) is called *polynomially decidable* whenever there is a deterministic Turing machine (DTM) which decides $\{(x, y_1 \ldots, y_k) : (x, y_1 \ldots, y_k) \in R\}$ in polynomial time; furthermore, $R$ is called *polynomially balanced* if there is a $k \in \mathbb{N}$ such that for all $(x, y_1 \ldots, y_k) \in R$: $|y_i| \leq |x|^k$ for all $i = 1, \ldots k$.

For a language $L$ and $k \geq 1$ the following holds: $L \in \Sigma_k^P$ if, and only if, there is a polynomially decidable and balanced $(k+1)$-ary relation $R$ such that $L = \{x \mid \exists y_1 \forall y_2 \exists y_3 \ldots Q y_k \ ((x, y_1 \ldots, y_k) \in R)\}$ where $Q = \forall$ (resp. $Q = \exists$) if $k$ is odd (resp. $k$ even) [Papadimitriou, 1994, Corollary 2 of Theorem 17.8].

**Theorem 10.29 (⇝ page 236).** *Let $\varphi$ be a level-$i$ formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \emptyset)$, $k \geq 1$, $i \geq 0$. Moreover, let $\mathfrak{M}$ be a* CGS*, and $q$ a state in $\mathfrak{M}$. Then, model checking $\mathfrak{M}, q \models \varphi$ can be done in time $\mathbf{\Delta^P_{i+2k+1}}$.*

*Proof.* By induction over $k$. In the following we restrict ourselves to $(\textbf{set-pl}\ \cdot)$ without loss of generality.

**Case** $k = 1$. Let $\varphi$ be a level-$i$ $\mathcal{L}^1_{ATLP}$ formula, $(\textbf{set-pl}\ \omega)$ an operator occurring in $\varphi$ such that $l(\{(\textbf{set-pl}\ \omega)\}) = i$ and $\omega = \sigma.Q_1\sigma_1Q_2\sigma_2 \ldots Q_n\sigma_n\varphi'$ where

$$\varphi' \in \mathcal{L}^{\mathrm{base}}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{T}(\{\sigma, \sigma_1, \ldots, \sigma_n\})).$$

Note that $\mathfrak{M}^{s,s_1,\ldots,s_n}, q \models \varphi'$ can be checked in polynomial time since all constructible plausibility terms are rectangular and the representation is directly given (see Corollary 10.14). Moreover, let $q'$ denote the state in which $\omega$ is evaluated. W.l.o.g. we can assume that $\varphi$ has the following structure:

$$\varphi \equiv (\textbf{set-pl}\ \omega)\mathbf{Pl}\,\langle\!\langle A\rangle\!\rangle\Box \mathsf{yes}.$$

Now, $\varphi$ is true in $\mathfrak{M}$ and $q$ if and only if there is a plausible strategy $s_A$ for $A$ and *no* plausible strategy $t$ with $t|_A = s$ such that $\mathfrak{M}', q \models_{\mathsf{CTL}} \neg \mathsf{A}\Box\mathsf{yes}$ where $\mathfrak{M}'$ is the trimmed model of $\mathfrak{M}$ wrt $t$. In the following we neglect the complexity needed to verify whether $s_A$ is plausible since the method *beatable* also verifies this property and its complexity is at least as high (cf. proof of Proposition 10.4). Thus, $\varphi$ is true if, and only if

$$\exists s_A \neg \big(\exists t\ (t \in \widehat{[\![\omega]\!]}^{q'} \text{ and } R_{\models}(\mathfrak{M}, q, s_A, t, \Box\mathsf{yes})\big)\big)$$
$$\text{iff } \exists s_A \neg \big(\exists t Q_1 s_1 Q_2 s_2 \ldots Q_n s_n\ (\mathfrak{M}^{t,s_1,\ldots,s_n}, q' \models \varphi'$$
$$\text{and } R_{\models}(\mathfrak{M}, q, s_A, t, \Box\mathsf{yes})\big)\big)$$
$$\text{iff } \exists s_A \forall t \bar{Q}_1 s_1 \bar{Q}_2 s_2 \ldots \bar{Q}_n s_n\ (\mathfrak{M}^{t,s_1,\ldots,s_n}, q' \not\models \varphi' \text{ or } \neg R_{\models}(\mathfrak{M}, q, s_A, t, \Box\mathsf{yes}))$$

where $R_{\models}(\mathfrak{M}, q, s_A, t, \Box\mathsf{yes}) = \mathit{true}$ iff $t|_A = s_A$ and $\mathfrak{M}', q \models_{\mathsf{CTL}} \neg\mathsf{A}\Box\mathsf{yes}$ where $\mathfrak{M}'$ is the "trimmed" model of $\mathfrak{M}$ wrt $t$, and $\bar{Q}$ is the dual quantifier to $Q$.

The latter conditions can be verified in polynomial time. We consider the number of quantifier alternations. Subsequent strategies which are quantified by quantifiers of the same type can be treated together. The same holds if the sequence starts with existential quantifiers. These strategies can be guessed together with strategy $t$. A quantifier level of $l(\{(\mathbf{set\text{-}pl}\ \omega)\}) = i$ denotes that it is sufficient to alternatingly guess $i$ witnesses. We obtain the following structure:

$$\exists s_A \forall x_t \exists x_1 \forall x_2 \ldots Q x_i$$

where $Q = \exists$ (resp. $Q = \forall$) if $i$ is even (resp. odd). Where $x_i$ denotes a witness for a strategy or several strategies if guessing can be combined.
Thus, according to Remark B.3 checking whether $\varphi$ is satisfied can be determined in time $\Sigma_{i+2}$ and the complete model checking algorithm for level-$i$ $\mathcal{L}^1_{ATLP}$ formula can be performed in time $\mathbf{\Delta^P_{i+3}}$ (there can be polynomially many such constructs).

**Induction step: $k \mapsto k+1$ ($k > 1$).** Let $\varphi$ be a level-$i$ $\mathcal{L}^{k+1}_{ATLP}$ formula and let $\omega$ be a term in $\varphi$ of the form $\omega = \sigma_1.Q_1\sigma_1 Q_2\sigma_2 \ldots Q_n\sigma_n\varphi'$ such that $l((\mathbf{set\text{-}pl}\ \omega)) = i$. Furthermore, let $\mathrm{RALT}(Q_1 \ldots Q_n) = j$; then, $l_{\varphi'} := ql(\mathcal{UO}(\varphi')) = i - j$ and $\varphi'$ is an $\mathcal{L}^k_{ATLP}$ formula. Thus, by induction hypothesis we have that $\varphi'$ can be model checked in time

$$\mathbf{\Delta^P_{r+1}} \quad \text{where } r := l_{\varphi'} + 2k.$$

Again, w.l.o.g. we can assume that $\varphi$ has the following structure:

$$\varphi \equiv (\mathbf{set\text{-}pl}\ \omega)\mathbf{Pl}\,\langle\!\langle A \rangle\!\rangle \Box \mathsf{yes}.$$

We proceed as in case $k = 1$. Firstly, a profile $s$ is guessed, then a profile $t$ and it is checked whether $t$ is plausible and coincides with $s$ wrt $A$ and whether the trimmed model (wrt $t$) satisfies $\neg\mathsf{A}\Box\mathsf{yes}$. We obtain the following structure:

$$\exists s_A \neg \left( \exists t\ (t \in \widehat{[\![\omega]\!]}^{q'} \text{ and } R_{\models}(\mathfrak{M}, q, s_A, t, \Box\mathsf{yes})) \right)$$

$$\text{iff } \exists s_A \neg \Bigg( \exists t Q_1 s_1 Q_2 s_2 \ldots Q_n s_n\ (\underbrace{\mathfrak{M}^{t,s_1,\ldots,s_n}, q' \models \varphi'}_{\in \mathbf{\Delta^P_{r+1}}}$$

$$\text{and } \underbrace{R_{\models}(\mathfrak{M}, q, s_A, t, \Box\mathsf{yes}))}_{\in \mathbf{P}} \Bigg).$$

Since $\mathfrak{M}^{t,s_1,\ldots,s_n}, q' \models \varphi'$ is invoked by a nondeterministic polynomial Turing machine we can assume that its model checking problem can be solved in $\mathbf{\Sigma_r^P}$ or $\mathbf{\Pi_r^P}$ instead of $\mathbf{\Delta_{r+1}^P}$; the polynomial effort of the deterministic machine can also be done by the invoking nondeterministic machine. Because $\text{RALT}(Q_1Q_2\dots Q_n) = j$ analogously to the case $k = 1$ we obtain that we can solve the problem in

$$\mathbf{P}^{\mathbf{\Sigma_{j+2}^P}^{\mathbf{\Sigma_r^P}}} = \mathbf{\Delta_{i+2(k+1)+1}^P}$$

as $j + 2 + r = j + 2 + (i - j) + 2k = i + 2(k + 1)$.

■

**Theorem 10.30 (⇝ page 236).** *Let $\varphi$ be a level-i formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\text{gt}, \Pi, \Omega)$, $\mathfrak{M}$ a well-behaved* CGSP*, and $q$ a state in $\mathfrak{M}$. Model checking $\mathfrak{M}, q \models \varphi$ can be done in $\mathbf{\Delta_{i+2(k+1)+1}^P}$.*

*Proof.* The proof is similar to the one of Theorem 10.29. In comparison to the claim of Theorem 10.29, $2k$ has changed to $2(k+1)$. The reason for this is that the final nesting (i.e. formulae in $\mathcal{L}^{\text{base}}_{ATLP}$) might contain hard-wired terms and it can not be verified in polynomial time anymore. This causes the change from $k$ to $k+1$ (it requires to guess $s_A$ and verify it against all responses $t$).■

**Proposition 10.32 (⇝ page 238).** *Let $\varphi$ be a* QSAT *instance. Then it holds that $\varphi$ is satisfiable if, and only if, $\mathfrak{M}_\varphi, q_0 \models \theta_\varphi$ where $\mathfrak{M}_\varphi$ and $\theta_\varphi$ are effectively constructible from $\varphi$ in polynomial time with respect to the length of the formula $\varphi$.*

*Proof.* Let $\varphi$ be a QSAT instance. We use the construction above (cf. page 238) to obtain $\mathfrak{M}'_\varphi$ and $\theta_\varphi$ where $\mathit{uniform}^i_{\mathbf{ATLP}}(\sigma)$ is obtained as follows: Firstly, we take the **ATLP+K** formula $\mathit{uniform}(\sigma|_i)$ (where $\sigma|_i$ refers to agent $i$'s strategy in $\sigma$) as described in Appendix B.4.3; then, we use the polynomial translation to change knowledge to ability, yielding a pure **ATLI** formula. Finally, we use the polynomial translation from **ATLI** to **ATLP** given in Section 6.3 (Proof of Proposition 6.32) to obtain a pure **ATLP** formula $\mathit{uniform}^i_{\mathbf{ATLP}}(\sigma)$. Hence, the latter formula is true if agent $i$'s strategy contained in the complete profile $\sigma$ is a uniform strategy. This shows that $\theta_\varphi$ can be constructed in polynomial time.

Model $\mathfrak{M}_\varphi$ is obtained from $\mathfrak{M}'_\varphi$ by the same scheme. Firstly, the construction from [Jamroga, 2007] referred to in Appendix B.4.3 is applied. Secondly, the resulting CGS with intentions is transformed to a CGSP using the construction from Section 6.3 (Proposition 6.32) again. The constructed model $\mathfrak{M}_\varphi$ is also polynomial with respect to $\varphi$.

We get that $\varphi$ is satisfiable if, and only if, $\mathfrak{M}_\varphi, q_0 \models \theta_\varphi$. ■

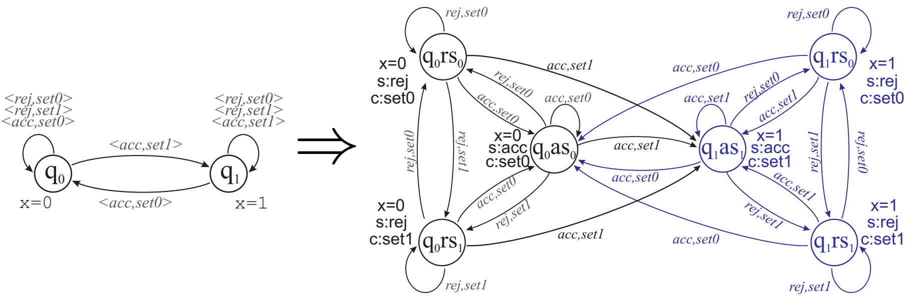


**Fig. B.2.** Memorising the last action profile in a simple 2-agent system.

### B.4.3 From $\mathbf{ATL}_{\mathbf{ir}}^{u}$ to ATLP with ATLI-Based Plausibility Terms

The reduction of $\mathbf{ATL}_{\mathbf{ir}}^{u}$ model checking to model checking of $\mathbf{ATLP}^{\mathbf{ATLI}}$ in "pure" CGSs is rather sophisticated. We do not present a reduction for full model checking of $\mathbf{ATL}_{\mathbf{ir}}^{u}$; it is enough to show the reduction for the kind of models that we obtain in Appendix B.3 (i.e., turn-based models with two agents, two "final" states $q_\top, q_\bot$, no cycles except for the loops at final states, and uncertainty appearing only in states one step before the end of the game, cf. Figure B.1).

Firstly, we reconstruct the concurrent epistemic game structure $\mathfrak{M}_p$ from Section B.3 so that the last action profile is always "remembered" in the final states. Then, we show how uniformity of strategies can be characterised with a formula of **ATLI** extended with epistemic operators. Thirdly, we show how the model and the formula can be transformed to remove epistemic links and operators (yielding a "pure" CGS and a formula of "pure" **ATLI**). Finally, we show how the resulting characterisation of uniformity can be "plugged" into an **ATLP** formula to require that only uniform strategy profiles are taken into account.

*Adding More Final States to the Model.*

We recall that the input of $\mathbf{ATL}_{\mathbf{ir}}^{u}$ model checking consists in our case of a concurrent epistemic game structure $\mathfrak{M}_p$ (like the one in Figure B.1) and an $\mathbf{ATL}_{\mathbf{ir}}^{u}$ formula $\Phi_p$ (cf. Proposition B.2). We begin the reduction by reconstructing $\mathfrak{M}_p$ to $\mathfrak{M}'_p$ in which the last action profile is "remembered" in the final states. The idea is based on the construction from [Goranko and Jamroga, 2004, Proposition 16] where it is applied to all states of the system, cf. Figure B.2.

In our case, we first create copies of states $q_\top, q_\bot$, one per incoming transition. That is, the construction yields states of the form $\langle q, \alpha_1, \ldots, \alpha_k \rangle$, where $q \in \{q_\top, q_\bot\}$ is a final state of the original model $\mathfrak{M}_p$, and $\langle \alpha_1, \ldots, \alpha_k \rangle$

is the action profile executed just before the system proceeded to $q$. Each copy has the same valuation of propositions as the original state $q$, i.e., $\pi'(\langle q, \alpha_1, \ldots, \alpha_k \rangle) = \pi(q)$. Then, for each action $\alpha \in Act$ and agent $i \in \mathbb{A}\mathrm{gt}$, we add a new proposition $\mathsf{i} : \alpha$. Moreover, we fix the valuation of $\mathsf{i} : \alpha$ in $\mathfrak{M}'_p$ so that it holds exactly in the final states which can be achieved by an action profile in which $i$ executes $\alpha$ (i.e., states $\langle q, \alpha_1, ..., \alpha_i, ..., \alpha_k \rangle$). We note that the number of both states and transitions in $\mathfrak{M}'_p$ is linear in the transitions of $\mathfrak{M}_p$. The transformation produces model $\mathfrak{M}'_p$ which is equivalent to $\mathfrak{M}_p$ in the following sense. Let $\varphi$ be a formula of $\mathbf{ATL}^u_{\mathbf{ir}}$ that does not involve special propositions $\mathsf{i} : \alpha$. Then, for all $q \in Q$: $\mathfrak{M}_p, q \models_{\mathbf{ATL}^u_{\mathbf{ir}}} \varphi$ iff $\mathfrak{M}'_p, q \models_{\mathbf{ATL}^u_{\mathbf{ir}}} \varphi$.

In $\mathfrak{M}'_p$, agents can "recall" their actions executed at states that involved some uncertainty (i.e., states in which the image of some indistinguishability relation $\sim_i$ was not a singleton). Now we can use **ATLI** (with additional help of knowledge operators, see below) to characterise uniformity of strategies.

*Characterising Uniformity in* **ATLI+K**.

We will now show that uniformity of a strategy can be characterised in **ATLI** *extended with epistemic operators* $K_a$ (that we call **ATLI+K**). $K_a\varphi$ reads as "agent $a$ knows that $\varphi$". The semantics of **ATLI+K** extends that of **ATLI** by adding the standard semantic clause from epistemic logic:

$$\mathfrak{M}, q \models K_a\varphi \text{ iff } \mathfrak{M}, q' \models \varphi \text{ for every } q' \text{ such that } q \sim_a q'.$$

We note that **ATLI+K** can be also seen as **ATEL** [van der Hoek and Wooldridge, 2002] extended with intentions.

Let us now consider the following formula of **ATLI+K**:

$$\mathit{uniform}(\sigma) \equiv (\mathbf{str}\sigma)\langle\langle\emptyset\rangle\rangle\Box \bigwedge_{i\in\mathbb{A}\mathrm{gt}} \bigvee_{\alpha\in d(i,q)} K_i\langle\langle\emptyset\rangle\rangle \bigcirc \mathsf{i} : \alpha.$$

The reading of $\mathit{uniform}(\sigma)$ is: suppose that profile $\sigma$ is played ($\mathbf{str}\sigma$); then, for all reachable states ($\langle\langle\emptyset\rangle\rangle\Box$), every agent has an action ($\bigwedge_{i\in\mathbb{A}\mathrm{gt}} \bigvee_{\alpha\in d(i,q)}$) that is determined for execution ($\langle\langle\emptyset\rangle\rangle \bigcirc \mathsf{i} : \alpha$) in every state indistinguishable from the current state ($K_i$). Thus, formula $\mathit{uniform}(\sigma)$ characterises the *uniformity* of strategy profile $\sigma$. Formally, for every concurrent epistemic game structure $\mathfrak{M}$, we have that $\mathfrak{M}, q \models_{\mathbf{ATLI+K}} \mathit{uniform}(\sigma)$ iff $[\sigma[a]]$ is uniform for each agent $a \in \mathbb{A}\mathrm{gt}$ (for all states reachable from $q$). Of course, only reachable states matter when we look for strategies that should enforce a temporal goal.

Note that the epistemic operator $K_a$ refers to incomplete information, but $\sigma$ is now an arbitrary (i.e., not necessarily uniform) strategy profile. We observe that the length of the formula is linear in the number of agents and actions in the model.

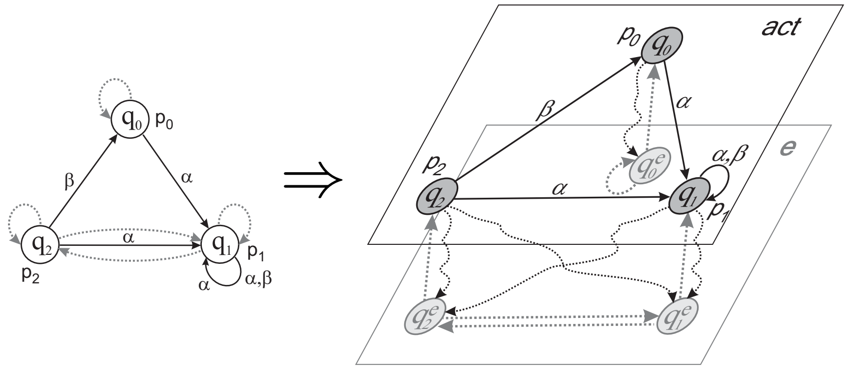



**Fig. B.3.** Getting rid of knowledge and epistemic links.

*Translating Knowledge to Ability.*

To remove the epistemic operators from formula $uniform(\sigma)$ and epistemic relations from model $\mathfrak{M}'_p$, we use the construction from [Jamroga, 2007] (which refines that from [Goranko and Jamroga, 2004, Section 4.4]). The construction yields a concurrent game structure $tr(\mathfrak{M}'_p)$ and an **ATLI** formula $tr(uniform(\sigma))$ with the following characteristics. The idea can be sketched as follows. The set of agents is extended with *epistemic agents* $e_i$ (one per $a_i \in \mathbb{A}\mathrm{gt}$), yielding $\mathbb{A}\mathrm{gt}'' = \mathbb{A}\mathrm{gt} \cup \mathbb{A}\mathrm{gt}^e$. Similarly, the set of states is augmented with *epistemic states* $q^e$ for every $q \in Q'$ and $e \in \mathbb{A}\mathrm{gt}^e$; the states "governed" by the epistemic agent $e_a$ are labeled with a special proposition $\mathsf{e_a}$. The "real" states $q$ from the original model are called "action" states, and are labeled with another special proposition $\mathsf{act}$. Epistemic agent $e_a$ can enforce transitions to states that are indistinguishable for agent $a$ (see Figure B.3 for an example).[3] Then, "$a$ knows $\varphi$" can be rephrased as "$e_a$ can only affect transitions to epistemic states where $\varphi$ holds". With some additional tricks to ensure the right interplay between actions of epistemic agents, we get the following translation of formulae:

[3] The interested reader is referred to [Jamroga, 2007] for the technical details of the construction.

$$\begin{aligned}
tr(p) &= \mathsf{p}, \qquad \text{for } \mathsf{p} \in \Pi \\
tr(\neg\varphi) &= \neg tr(\varphi), \\
tr(\varphi \vee \psi) &= tr(\varphi) \vee tr(\psi), \\
tr(\langle\langle A\rangle\rangle \bigcirc \varphi) &= \langle\langle A \cup \mathbb{A}\mathrm{gt}^e\rangle\rangle \bigcirc (\mathsf{act} \wedge tr(\varphi)), \\
tr(\langle\langle A\rangle\rangle \Box \varphi) &= \langle\langle A \cup \mathbb{A}\mathrm{gt}^e\rangle\rangle \Box (\mathsf{act} \wedge tr(\varphi)), \\
tr(\langle\langle A\rangle\rangle \varphi \mathcal{U} \psi) &= \langle\langle A \cup \mathbb{A}\mathrm{gt}^e\rangle\rangle (\mathsf{act} \wedge tr(\varphi)) \mathcal{U} (\mathsf{act} \wedge tr(\psi)), \\
tr(K_i\varphi) &= \neg\langle\langle e_1, ..., e_i\rangle\rangle \bigcirc \big(\mathsf{e_i} \wedge \langle\langle e_1, ..., e_k\rangle\rangle \bigcirc (\mathsf{act} \wedge \neg tr(\varphi))\big).
\end{aligned}$$

Note that the length of $tr(\varphi)$ is linear in the length of $\varphi$ and the number of agents $k$. Two important facts follow from [Jamroga, 2007, Theorem 8]:

**Lemma B.4.** *For every* ICGS $\mathfrak{M}$ *and formula of* $\mathbf{ATL}_{\mathbf{ir}}^{u}$ *that does not include the special propositions* $\mathsf{act}, \mathsf{e_1}, \ldots, \mathsf{e_k}$*, we have*

$$\mathfrak{M}, q \models_{\mathbf{ATL}_{\mathbf{ir}}^{u}} \varphi \quad \textit{iff} \quad tr(\mathfrak{M}), q \models_{\mathbf{ATL}_{\mathbf{ir}}^{u}} tr(\varphi).$$

**Lemma B.5.** *For every* ICGS $\mathfrak{M}$*, we have*

$$\mathfrak{M}, q \models_{\mathbf{ATLI+K}} uniform(\sigma) \quad \textit{iff} \quad tr(\mathfrak{M}), q \models_{\mathbf{ATLI+K}} tr(uniform(\sigma)).$$

*Putting the Pieces Together: The Reduction.*

We observe that $\mathbf{ATL}_{\mathbf{ir}}^{u}$ can be seen as **ATL** where only uniform strategy profiles are allowed. An **ATLI** formula that characterises uniformity has been defined in the previous paragraphs. It can be now plugged into our "**ATL** with Plausibility" to restrict agents' behaviour in the way the semantics of $\mathbf{ATL}_{\mathbf{ir}}^{u}$ does. This way, we obtain a reduction of $\mathrm{SNSAT}_2$ to model checking of $\mathbf{ATLP}^{\mathbf{ATLI}}$.

**Proposition B.6.**

$$z_p \quad \textit{iff} \quad tr(\mathfrak{M}'_p), q_0^p \models_{\mathbf{ATLP}^{\mathbf{ATLI}}} (\textbf{\textit{set-pl}}\, \sigma.tr(uniform(\sigma)))\mathbf{Pl}\, tr(\Phi_p).$$

*Proof.* We have $z_p$ iff $\mathfrak{M}'_p, q_0^p \models_{\mathbf{ATL}_{\mathbf{ir}}^{u}} \Phi_p$ iff $tr(\mathfrak{M}'_p), q_0^p \models_{\mathbf{ATL}_{\mathbf{ir}}^{u}} tr(\Phi_p)$ iff $tr(\mathfrak{M}'_p), q_0^p \models_{\mathbf{ATLP}^{\mathbf{ATLI}}} (\textbf{set-pl}\ \sigma.tr(uniform(\sigma)))\mathbf{Pl}\, tr(\Phi_p)$. The details follow similarly to Theorems 10.20 and 10.17 and Proposition 10.19. ■

## B.5 Model Checking CoalATL

**Lemma 10.47 (⇝ page 247).** *Let* $\varphi(X)$ *be a Boolean formula in positive normal form.*

(a) *If $T$ is a satisfying truth assignment of $\varphi$, then there is a $\mathbf{v}$-choice of $\mathfrak{M}(\varphi)$ such that for the set $L$ of all literal states reachable from $q_0$ it holds that $\{x \in X \mid q_w \in L$ and $\chi_w = x\} \subseteq T$ and $\{x \in X \mid q_w \in L$ and $\chi_w = \neg x\} \cap T = \emptyset$.*

(b) *If there is a $\mathbf{v}$-choice of $\mathfrak{M}(\varphi)$ such that for the set $L$ of all literal states reachable from $q_0$ we have that for any $q_w, q_v \in L$ the formula $\chi_w \wedge \chi_v$ is satisfiable (i.e. there are no complementary literals) then the set $\{x \in X \mid q_w \in L$ and $\chi_w = x\}$ is a satisfying truth assignment of $\varphi$.*

*Proof.*

(a) Let $T$ be a satisfying truth assignment of $\varphi$. We construct a $\mathbf{v}$-choice with the stated property. Firstly, we call a state $q_w$ a $\mathbf{v}$-state (resp. $\mathbf{r}$-state) if $q_w$ is controlled by $\mathbf{v}$ (resp. $\mathbf{r}$). Next, we consider a state $q_w$ such that $q_{w1}$ and $q_{w2}$ are literal states. Firstly, let us consider the case that $q_w$ is a $\mathbf{v}$-state. Then, we label it with $\top$ if for some $i$ we have that $x \in T$ (resp. $x \notin T$) if $\chi_{w_i} = x$ (resp. $\chi_{w_i} = \neg x$) for $x \in X$; otherwise we label the state with $\bot$. Secondly, if $q_w$ is an $\mathbf{r}$-state we label it with $\top$ if the above condition holds for *both* $i$; otherwise with $\bot$.
In the second step, we label each $\mathbf{v}$-state with two labelled successor states with $\top$ if at least one of its successors has label $\top$; otherwise we assign to it the label $\bot$. On the other hand, if it is an $\mathbf{r}$-state, if both successors have the label $\top$ we label it $\top$; otherwise we assign to it label $\bot$. We proceed like this until all states $q_w$ with a relevant $w$ are labelled.
Now, in lexicographical order we go through all $w$ relevant for $\varphi$ and perform the following steps. If $q_w$ is a $\mathbf{v}$-state reachable from $q_0$, we remove the transition to the successor which is labelled $\bot$ and if both transitions are labeled $\top$ we remove any of its transitions. Else, if $q_w$ is a $\mathbf{v}$-state and *not* reachable from $q_0$ we remove any of its transitions. Following this procedure, we have to show that there is no reachable $\mathbf{v}$-state such that both successors are labelled $\bot$. Suppose such a state $q_w$ is reachable. Since the verifier has no strategy to avoid this state (note that $w$ is lexicographically minimal) the disjunction $\chi_{w1} \vee \chi_{w2}$ must be true under $T$ in order to make $\varphi$ true. However, since both states $q_{w1}$ and $q_{w2}$ are labelled $\bot$ the truth assignment $T$ falsifies $\chi_{w1}$ and $\chi_{w2}$ (that can easily be seen by induction). Contradiction!
Hence, the construction yields a $\mathbf{v}$-choice. We need to show that there is a $\mathbf{v}$-choice such that $S := \{x \in X \mid q_w \in L \text{ and } \chi_w = x\}$ is a subset of $T$. Firstly, assume there is some $x \in S$ not in $T$. Let $q_{wi}$ be some reachable literal state with $\chi_{wi} = x$. By construction of the $\mathbf{v}$-choice, this implies that the state $q_w$ is an $\mathbf{r}$-state. (Otherwise, the other alternative would have been selected (according to the labelling algorithm); or, both successors of $q_w$ would be labelled $\bot$ what is not possible as shown above.) But this

means, that the formula $\chi_{w1} \wedge \chi_{w2}$ needs to be satisfied and hence $x \in T$. Secondly, assume that $x \in \{x \in X \mid q_w \in L \text{ and } \chi_w = \neg x\} \cap T$ and let $q_{wi}$ be the state reachable with $\chi_{wi} = \neg x$. Following the same reasoning as above the state $q_w$ must be an **r**-state and this contradicts the fact that the formula $\chi_{w1} \wedge \chi_{w2}$ must be true under $T$.

(b) Let us consider a **v**-choice $v$ of $\mathfrak{M}(\varphi)$ with the stated property and suppose that $T := \{x \in X \mid q_w \in L \text{ and } \chi_w = x\}$ is not a satisfying truth assignment of $\varphi$. Note that the **v**-choice corresponds to the selection of the left (resp. right) hand subformula of any subformula $\chi_w$ with $lc(\chi_w) = \vee$. We say that a subformula $\chi_w$ is reachable if the state $q_w$ is reachable in the very **v**-choice $v$. Since $T$ is not a satisfying truth assignment there is some reachable subformula (possibly $\varphi$ itself) not satisfied by $T$. Let $\chi_w$ be such a reachable subformula with lexicographically maximal $w$ (so, it is a relative "small" subformula). We consider two cases. Firstly, suppose that $\chi_w$ is not satisfiable. Then, due to the maximality of $w$, $\chi_w$ must be a conjunction of literals among that are two complementary ones, we denote them by $\chi_v$ and $\chi_{v'}$ (otherwise it would be satisfiable); hence, we have that $q_v, q_{v'} \in L$. Contradiction!
So, suppose $\chi_w$ is satisfiable but false under $T$. Then, due to the maximality of $w$, $\chi_w$ can either be $x_i$ for some $x_i \notin T$ or be $\neg x_j$ for some $x_j \in T$. Suppose $\chi_w = x_i$; then $q_w$ is reachable and thus $x_i \in T$. Contradiction! On the other hand, if $\chi_w = \neg x_j$ then $q_w \in L$ and since $x_j \in T$ there is some other state $q_v \in L$ with $\chi_v = x_j$ and $\chi_w \wedge \chi_v$ not satisfiable. Contradiction! ■

**Proposition 10.48 (⇝ page 248).** *The model $\mathfrak{M}(\varphi)$ is constructible in polynomial-time wrt the size of $\varphi$ and we have that*

$$\varphi(X) \textit{ is satisfiable if, and only if, } \mathfrak{M}(\varphi), q_0 \models \langle\!|\mathbf{v}|\!\rangle \Diamond \mathsf{sat}.$$

*Proof.* Firstly, we analyse the construction to show that its length is polynomial wrt the input. The number of agents is polynomial in $X$; the number of states is polynomial in the number of subformulae of $\varphi$ and for each state there are at most two transitions (apart form the states $q_\bot$ and $q_\top$).

"$\Rightarrow$": Let $\varphi(X)$ be satisfiable and let $T \subseteq X$ be a satisfying truth assignment. By construction of $\mathfrak{M}(\varphi)$ we have that $C := \{\mathbf{v}, a_i, \bar{a}_j \mid x_i \in T, x_j \notin T\} \in \mathfrak{sem}(\zeta(\{\mathbf{v}\})(q_0))$; hence, it suffices to show that $\mathfrak{M}(\varphi), q_0 \models \langle\!\langle C \rangle\!\rangle \Diamond \mathsf{sat}$. Now by Lemma 10.47(a) there is a strategy $s_\mathbf{v}$ of $\mathbf{v}$ such that for the set $L$ of all literal states reachable from $q_0$ we have that $S := \{x \mid q_w \in L \text{ and } \chi_w = x\} \subseteq T$ and $\{x \mid q_w \in L \text{ and } \chi_w = \neg x\} \cap T = \emptyset$. Hence, for any $x_i \in S$ the agent $a_i$ is in $C$ and for any $x_i \in \{x \mid q_w \in L \text{ and } \chi_w = \neg x\}$ the agent $\bar{a}_i$ is in $C$. Finally, the strategy $s_a$ of $a \in C \backslash \{\mathbf{v}\}$ is to execute $\top$ if $a = a_i$ and $\bot$ if $a = \bar{a}_i$ in the states controlled by $a$. The complete strategy profile consisting of $s_b$ for $b \in C$ ensures $\Diamond \mathsf{sat}$.

"⇐": Suppose that $\mathfrak{M}(\varphi), q_0 \models \langle\!|\mathbf{v}|\!\rangle\Diamond\mathsf{sat}$. Then there is a coalition $C \in \mathfrak{sem}(\zeta(\{\mathbf{v}\})(q_0))$ such that $\mathfrak{M}(\varphi), q_0 \models \langle\!\langle C\rangle\!\rangle\Diamond\mathsf{sat}$. By Lemma 10.47(b) it remains to show that for the set $L$ of all literal states reachable from $q_0$ we have that for any $q_w, q_v \in L$ the formula $\chi_w \wedge \chi_v$ is satisfiable. Suppose the contrary; that is, there are $q_w, q_v \in L$ such that $\chi_w \wedge \chi_v \equiv \bot$; say $\chi_w = x_i$. By Proposition 10.44, $C \backslash \{\mathbf{v}\}$ does correspond to a truth assignment of $\varphi$; hence, either $a_i \in C$ or $\bar{a}_i \in C$. Hence, if $\bar{a}_i \in C$ then $a_i \notin C$ and the opponents (to whom $a_i$ belongs) have a strategy to reach $q_\bot$ from $q_w$. But this contradicts $\mathfrak{M}(\varphi), q_0 \models \langle\!\langle C\rangle\!\rangle\Diamond\mathsf{sat}$. The same reasoning is applied if $a_i \in C$ and $\bar{a}_i \notin C$. ■

**Lemma 10.52 (⇝ page 250).** *Let $I = (\varphi_1, \ldots, \varphi_p)$ be an* SNSAT$_1$ *instance.*

*(a) Let $T = (T_1, \ldots, T_p)$ be a solution for $I$. For all $r = 1, \ldots, p$, if $z_r \in T_r$ then there is a* **v***-choice of $\mathfrak{M}(\varphi_r)$ such that for the set $L$ of all literal states reachable from $q_0^r$ and which belong to $\mathfrak{M}(\varphi_r)$ it holds that $\{x \in X^r \cup Z \mid q_w \in L$ and $\chi_w^{\varphi_r} = x\} \subseteq T_r$ and $\{x \in X^r \cup Z \mid q_w \in L$ and $\chi_w^{\varphi_r} = \neg x\} \cap T_r = \emptyset$.*

*(b) Let $I^{p-1} = (\varphi_1, \ldots, \varphi_{p-1})$ and let $T^{p-1} = (T_1^{p-1}, \ldots, T_{p-1}^{p-1})$ be an $I^{p-1}$-witness. Then, if there is a* **v***-choice of $\mathfrak{M}(\varphi_p)$ such that for the set $L$ of all literal states reachable from $q_0^p$ that belong to $\mathfrak{M}(\varphi_p)$ we have that*

*(i) for any $q_w, q_v \in L$ the literals $\chi_w^{\varphi_p}$ and $\chi_v^{\varphi_p}$ are non-complementary; and*

*(ii) if $q_v \in L$ with $\chi_v^{\varphi_p} = z_i$ (resp. $\chi_v^{\varphi_p} = \neg z_i$) then $z_i \in T_i$ (resp. $z_i \notin T_i$);*

*then, $T = (T_1^{p-1}, \ldots, T_{p-1}^{p-1}, T_p)$ is a solution for $I$ where*

$$T_p = \{x \in X^r \mid q_w \in L \text{ and } \chi_w = x\} \cup \{z^i \mid z^i \in T_i^{p-1},\ i < p\} \cup \{z^p\}.$$

*(c) $\mathfrak{M}(I), q_0^i \models \varphi_i$ if, and only if, $\mathfrak{M}(I), q_0^i \models \varphi_j$; and $\mathfrak{M}(I), q_0^i \models \neg\varphi_i$ if, and only if, $\mathfrak{M}(I), q_0^i \models \neg\varphi_j$ for all $j \geq i$.*

*Proof.*

(a) Let $z_r \in T_r$. As in the proof of Lemma 10.47(a) we proceed with the following labelling: We consider a state $q_w$ such that $q_{w1}$ and $q_{w2}$ are literal states in $\mathfrak{M}(\varphi_p)$. Firstly, let us consider the case that $q_w$ is a **v**-state. Then, we label it with $\top$ if for some $i$ with $\chi_{w_i} = x \in X^r \cup Z$ (resp. $\chi_{w_i} = \neg x$, $x \in X^r \cup Z$) we have that $x \in T_r$ (resp. $x \notin T_r$); otherwise we label the state with $\bot$. Secondly, if $q_w$ is a **r**-state we label it with $\top$ if the above condition holds for *both* $i$; otherwise by $\bot$. In the second step, (labelling non-literal states) we proceed exactly as before.
In the second step, we label each **v**-state with two labelled successor states with $\top$ if at least one of its successors has the label $\top$; otherwise we assign to it the label $\bot$. On the other hand, if it is a **r**-state, if both successors have the label $\top$ we label it $\top$; otherwise we assign to it the label $\bot$. We proceed like that until all states $q_w$ with a relevant $w$ are labelled.

The **v**-choice is constructed in the very same way as in Lemma 10.47(a) and also the verification that the **v**-choice has the same properties is done in the very same way.

(b) Let us consider a **v**-choice with the stated properties (i) and (ii). We adopt the notation from the proof of Lemma 10.47(b). By definition of $T_p$ it is obvious that it satisfies conditions 1 and 3 of Definition 10.50.
Suppose that $T$ is not a solution of $I$; that is, condition 2 of Definition 10.50 is violated. Since $T^{p-1}$ is a witness of $I^{p-1}$ there is a reachable subformula $\chi_w$ of $\varphi_p$ with lexicographically minimal $w$ that is not satisfiable given the valuation of the $z^{i}$'according to $T^{p-1}$ (note that the $z_i$'s are uniquely determined in $T^{p-1}$). Note, that the condition (ii) guarantees that the "right" value is chosen for $z_i \in T_i$ and $z_i \notin T_i$. The rest of the proof is done analogously to the one given in Lemma 10.47(b) by considering the literal states corresponding to variables $X^p \cup Z$. This proves that $T$ is a solution.

(c) Along each path from $q_0^i$ to a state labelled $\mathsf{sat}$ there are at most $i-1$ states labelled $\mathsf{neg}$. Hence, the truth of a formula $\varphi_j$ with $j \geq i$ is equivalent to the truth of $\varphi_i$.

■

**Theorem 10.53 (⇝ page 251).** *The size of $\mathfrak{M}(I)$ and of the formulae $\varphi_p$ is polynomially in the size of the* SNSAT$_1$ *instance $I = (\varphi_1, \ldots, \varphi_p)$ and we have the following:*

*There is a solution $T = (T_1, \ldots, T_r)$ of $I^r = (\varphi_1, \ldots, \varphi_r)$ if, and only if,*
$$\mathfrak{M}(I^r), q_0^r \models \varphi_l$$

*for $l \geq r$ and $r \leq p$.*

*Proof.* That $\mathfrak{M}(I)$ is polynomially wrt $\varphi$ follows from Proposition 10.48 and by the way it is constructed. The size of $\varphi_p$ also is polynomially in $\varphi(X)$.

The proof is done by induction on $r$.

*Induction start.* The induction starts with $r = 1$. An SNSAT$_1$-instance with $r = 1$ is given by $\exists Y_1 \varphi(X^1)$. This corresponds to the satisfiability problem. So, this case is proven in Proposition 10.48 and Lemma 10.52(c).

*Induction step.* Suppose the claim holds up to $r < p$.We show that there is a solution $T = (T_1, \ldots, T_{r+1})$ of $I^{r+1}$ iff $\mathfrak{M}(I^{r+1}), q_0^{r+1} \models \varphi_{l+1}$.

"⇒": Let $T = (T_1, \ldots, T_{r+1})$ be a solution for $I^{r+1}$.

From the reduction-suitableness we have that there is a coalition $C \in \mathfrak{sem}(\zeta(\{\mathbf{v}\})(q_0^p))$ with

$$C = \{\mathbf{v}\} \cup \bigcup_{i=1,\ldots,r+1} \{a_j^i \mid a_j^i \in T_i\} \cup \bigcup_{i=1,\ldots,r+1} \{\bar{a}_j^i \mid x_j^i \notin T_i\}.$$

Note that, if $z_i \in T_i$ then $T_i = \{x_j^i \mid a_i^j \in C,\ j = 1,\ldots,s\} \cup \{z_j \mid \mathfrak{M}(I^{r+1}), q_0^j \models \varphi_j,\ j = 1,\ldots,i\}$.

We show that $\mathfrak{M}(I^{r+1}), q_0^{r+1} \models \langle\!\langle C\rangle\!\rangle(\neg\mathsf{neg}\,\mathcal{U}(\mathsf{sat}\vee(\mathsf{neg}\wedge\langle\!\langle\emptyset\rangle\!\rangle\bigcirc\neg\varphi_{i+1}))$. Let $s_{\mathbf{v}}^{r+1}$ be the partial strategy corresponding to the $\mathbf{v}$-choice according to Lemma 10.52(a). The set of reachable literal states $L^{r+1}$ in $\mathfrak{M}(\varphi_{r+1})$ corresponds to finite sequences of states starting in $q_0^{r+1}$. In the following, we call such literal states referring to some $z^i$ (resp. $\neg z^i$), *$z$ (resp. $\neg z$)-literal states* and the others *$x$ (resp. $\neg x$)-literal states.* Let $\lambda$ be a *finite* sequence of states starting in $q_0^{r+1}$ and ending in one of the states in $L^{r+1}$. Then this sequence has one of the following properties: (1) The last state is an $x$, $z$, $\neg x$-literal state and there is no $\neg z$-literal state on it; or (2) the last state is a $\neg z$-literal state and there is no other $\neg z$-literal state on it. Note, that due to Lemma 10.52(a) it is not possible to reach two complementary literal states. We show, that we can extend $s_{\mathbf{v}}$ to a strategy $s_C$ that witnesses the truth of $\varphi_{i+1}$.

Case 1. For a path ending with an $x$-literal state $\chi_w = x_i^{r+1}$ we have that $x_i^{r+1} \in T_{r+1}$ and thus $a_i^{r+1} \in C$. If this agent executes $\top$ the next state is $q_\top$.

Case 2. For a path ending with an $\neg x$-literal state $\chi_w = \neg x_i^{r+1}$ we have that $x_i^{r+1} \notin T_{r+1}$ and thus $\bar{a}_i^{r+1} \in C$. If this agent executes $\bot$ the next state is $q_\top$.

Case 3. For a path ending with an $z$-literal state $q_w$, $\chi_w = z^i$, we have that $z^i \in T_{r+1}$ hence also $z^i \in T_i$. By induction hypothesis we have that $\mathfrak{M}(I), q_0^i \models \varphi_r$. Now, it is easily seen by a further induction that the very same coalition $C$ has a winning strategy $s'_C$ witnessing $\varphi_r$. We combine the strategy $s_{\mathbf{v}}$ with $s'_C$.

Case 4. For a path ending with an $\neg z$-literal state $q_w$, $\chi_w = \neg z^i$, we have that $z^i \notin T_{r+1}$ hence also $z^i \notin T_i$. By induction hypothesis we have that $\mathfrak{M}(I), q_0^i \not\models \varphi_r$; hence, $\mathfrak{M}(I^{r+1}), q_w \models \mathsf{neg} \wedge \langle\!\langle\emptyset\rangle\!\rangle \bigcirc \neg\varphi_l$.

These cases provide us with the desired strategy $s_C$ witnessing $\varphi_{l+1}$ for the chosen coalition $C$.

"$\Leftarrow$": Suppose that $\mathfrak{M}(I^{r+1}), q_0^{r+1} \models \varphi_{l+1}$.

Let $z^i$ with $i < r+1$ and with maximal index $i$ be such that $\mathfrak{M}(I^i), q_0^i \models \varphi_i$. Then, there is a solution $T^i = (T_1,\ldots,T_i)$ of $I^i$ by induction hypothesis. According to Proposition 10.51(b) and (c) we can extend this solution to an $I^r$ witness $T^r = (T_1,\ldots,T_r)$ of $I^r$.

Now, since $\mathfrak{M}(I^{r+1}), q_0^{r+1} \models \varphi_{l+1}$, there is a coalition $C \in \mathfrak{sem}(\zeta(\mathbf{v})(q_0^{r+1}))$ and a strategy $s_C$ that witnesses the truth of the formula. Let $L$ be the set of all reachable literal states in $\mathfrak{M}(\varphi_{r+1})$ under $s_C$ (a $\mathbf{v}$-choice is contained implicitly). We show that the preconditions (i) and (ii) of Lemma 10.52(b) are satisfied by this $\mathbf{v}$-choice.

Condition (i). Firstly, for each reachable $x/\neg x$-literal state $q_w^{r+1} \in L$ with $\chi_w^{\varphi_{r+1}} = x$ we must have that $a_i^{r+1}$ (resp. $\bar{a}_i^{r+1}$) is in $C$ if $x = x_i^{r+1}$ (resp. $x = \neg x_i^{r+1}$) (otherwise the formula $\neg\mathsf{neg}\mathcal{U}\mathsf{sat}$ would not be true). Hence, there cannot be any complementary $x$-literal states contained in $L$ due to the reduction-suitable semantics (again, otherwise $\neg\mathsf{neg}\mathcal{U}\mathsf{sat}$ would not be true).

Secondly, observe that for any positive $z$-literal state $q_w^{r+1} \in L$ with $\chi_w^{\varphi_{r+1}} = z^i$ we have that $\mathfrak{M}(I^{r+1}), q_w^{r+1} \models \varphi_{l+1}$ and by Lemma 10.52(c) also $\mathfrak{M}(I^{r+1}), q_0^i \models \varphi_i$; hence, by induction hypothesis, $z^i \in T_i$.

For each negative $z$-literal state $q_w^{r+1} \in L$ with $\chi_w^{\varphi_{r+1}} = \neg z^i$ we have that $\mathfrak{M}(I^{r+1}), q_0^i \models \neg\varphi_l$ and thus, by Lemma 10.52(c) $\mathfrak{M}(I^{r+1}), q_0^i \models \neg\varphi_i$. By induction hypothesis, we have that for *any* $I^i$-witness $(T_1', \ldots, T_i')$ it holds that $z^i \notin T_i'$. Contradiction to $z^i \in T_i$.

Condition (ii). Let $q_v \in L^p$ with $\chi_v^{\varphi_p} = z_i$. We show that $z_i \in T_i$. Suppose the contrary. Then, $\mathfrak{M}(I^{r+1}), q_0^i \models \neg\varphi_i$ and by Lemma 10.52(c) also $\mathfrak{M}(I^{r+1}), q_0^i \models \neg\varphi_{l+1}$. However, since along the path from $q_0^{r+1}$ to $q_0^i$ we always have $\neg\mathsf{neg}$ the coalition $C$ performing strategy $s_C$ witnesses the formula $\varphi_{l+1}$ in $q_0^i$. Contradiction!

Secondly, assume $q_v \in L$ with $\chi_v^{\varphi_{r+1}} = \neg z_i$ and again, by the sake of contradiction, that $z_i \in T_i$. Then $\mathfrak{M}(I^{r+1}), q_0^i \models \varphi_i$ and thus also $\mathfrak{M}(I^{r+1}), q_0^i \models \varphi_l$ by Lemma 10.52(c). However, on the path to $q_v \in L$ we have visited a state labeled $\mathsf{neg}$; hence, we must have $\mathfrak{M}(I^{r+1}), q_0^i \models \neg\varphi_l$. Contradiction!

Now by Lemma 10.52(b), we can construct a solution for $I^{r+1}$ form $T^r$.

■

## B.6 Model Checking CL on Implicit Models

**Theorem 5.24 (⇝ page 89).** *Model checking* $\mathbf{CL_{IR}}$, $\mathbf{CL_{Ir}}$, $\mathbf{CL_{ir}}$, *and* $\mathbf{CL_{iR}}$ *over implicit* CGSs *is* $\mathbf{\Delta_3^P}$*-complete with respect to the size of the model and the length of the formula. Moreover, it is* $\mathbf{\Sigma_2^P}$*-complete for the "simple" variants of CL.*

*Proof.* The upper bound follows from the result that model checking **ATL** is in $\mathbf{\Delta_3^P}$.

We extend the proof from [Laroussinie et al., 2008] such that only the next time operator is used. The proof is done by reducing the $\mathbf{\Delta_3^P}$-complete problem SNSAT$_2$. A SNSAT$_2$ instance $\mathcal{I}$ consists of formulae

$$(\star) \qquad z_i = \exists X_i \forall Y_i \psi_i(z_1, \ldots, z_{i-1}, X_i, Y_i)$$

where $X_i = \{x_i^1, \ldots, x_i^s\}$ and $Y_i = \{y_i^1, \ldots, y_i^s\}$ are sets of variables and $s \in \mathbb{N}$ for $i = 1, \ldots, m$. Accordingly to the truth of the formulae $\psi_i$ the value of each $z_i$ is uniquely defined. A valuation of $\mathcal{I}$ is a mapping $v_\mathcal{I}$ assigning these unique values to each variable $z_i$. Moreover, if $v_\mathcal{I}(z_i) = \top$ we define $v_\mathcal{I}^{z_i} : X_i \to \{\top, \bot\}$ to be some valuation of the variables $X_i$ that witnesses the truth of $z_i$. Note, that each $z_i$ recursively depends on $z_{i-1}, \ldots, z_1$. In the following we will often omit the subscript $\mathcal{I}$.

We construct the following implicit CGS $\mathfrak{M}_\mathcal{I}$ for a given SNSAT$_2$ instance $\mathcal{I}$. Firstly, we introduce agents, each controlling one variable. There are agents $a_i^j$ (one agent per variable $x_i^j$) with actions $\{\top, \bot\}$, $b_i^j$ (one agent per variable $y_i^j$) with actions $\{\top, \bot\}$, $c_i$ (one agent per $z_i$) with actions $\{\top, \bot\}$, and $d$ (the "selector") with actions $\{1, \ldots, m\}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, s$. We use $A$ (resp. $C$ and $B$) to denote the set of all agents $a_i^j$ (resp. $c_i$ and $b_i^j$).

The states of the model are given by states $q_i$ and $\bar{q}_i$ (one per $z_i$) and the two states $q_\top, q_\bot$. States $\bar{q}_i$ are labelled with proposition $\mathsf{neg}$ and state $q_\top$ is labelled with $\mathsf{sat}$.

Before giving the formal definition of the encoded transition function, we explain the role of the agents. Agents $a_i^j$ (resp. $b_i^j$ and $c_i$) determine the value of the variables $x_i^j$ (resp. $y_i^j$ and $z_i$). Action $\top$ (resp. $\bot$) sets them true (resp. false). Agent $d$ has a more elaborated function. Once, all moves of the other agents are fixed, the agent can decide to "check" whether formula $\psi_i$ holds regarding the actions of the other agents by executing action $i$. If the check is successful, the system goes to the winning state $q_\top$. If not, it goes to the losing state $q_\bot$. However, there are some exceptions to that which will be presented in the formal definition of the encoded transition function.

The part $(\varphi_0^i, q_0^i), \ldots, (\varphi_{t_i}^i, q_{t_i}^i)$ in the encoded transition function associated with state $q_i$ is defined as follows (where $\psi_i'$ denotes the formula $\psi_i$ in ($\star$) in which each occurrence of $x_i^j$ (resp. $y_i^j$ and $z_i$) is replaced by $\mathsf{exec}_\top^{\mathsf{a}_i^j}$ (resp. $\mathsf{exec}_\top^{\mathsf{b}_i^j}$ and $\mathsf{exec}_\top^{\mathsf{c}_i}$) (recall, that $\mathsf{exec}_\alpha^{\mathsf{a}}$ means that agent $a$ executes action $\alpha$)):

$$(\mathsf{exec}_\mathsf{k}^\mathsf{d} \wedge (\wedge_{j=i-1,\ldots,k} \mathsf{exec}_\top^{\mathsf{c}_j}) \wedge \psi_k', q_\top)_{k=i,\ldots,1}, \tag{B.1}$$

$$(\mathsf{exec}_\mathsf{k}^\mathsf{d} \wedge (\wedge_{j=i-1,\ldots,k} \mathsf{exec}_\top^{\mathsf{c}_j}), q_\bot)_{k=i,\ldots,1}, \tag{B.2}$$

$$(\mathsf{exec}_\mathsf{k}^\mathsf{d} \wedge \neg \mathsf{exec}_\top^{\mathsf{c}_k}, \bar{q}_k)_{k=i-1,\ldots,1}, \tag{B.3}$$

$$(\top, q_\top) \tag{B.4}$$

Moreover, there are loops at states $q_\top$ and $q_\bot$ and transitions from $\bar{q}_i$ to $q_i$ for $i = 1, \ldots, m$. The following lemma is fundamental to our reduction.

**Lemma B.7.** *Let* $\chi_0 = \top$ *and*

$$\chi_{r+1} = [A \cup C](\mathsf{sat} \vee (\mathsf{neg} \wedge [\emptyset] \neg \chi_r))$$

*for* $r = 0, \ldots, m-1$ *where* $\mathsf{sat}$ *and* $\mathsf{neg}$ *are propositional symbols. Then, for all* $i \le m$ *and* $r \ge i$ *it holds that*

$$\mathfrak{M}_{\mathcal{I}}, q_i \models \chi_r \text{ iff } v_{\mathcal{I}}(z_i) = \top.$$

*Proof.* We proceed by induction on $i$. Firstly, we consider the base case $i = 1$.

"$\Rightarrow$": Suppose that $\mathfrak{M}, q_1 \models \chi_r$ for $r \ge 1$. Due to the definition of the transition function only rules (1,2,4) are present; hence, only $q_\top$ and $q_\perp$ are reachable. That is, the formula $\mathfrak{M}, q_1 \models [A \cup C]\mathsf{sat}$ must be satisfied (as the label $\mathsf{neg}$ cannot become true). But then, there must be a valuation of the $x_1^j$'s such that for all valuations of the $y_1^j$'s, $\psi_1$ evaluates true; hence, $v(z_1) = \top$.

"$\Leftarrow$": Suppose $v(z_1) = \top$. Then, there is a valuation of the variables $x_1^j$ such that for all valuations of $y_1^j$ the formula $\psi_1$ evaluates true. It is easily seen that the strategy in which each agent in $A$ plays according to the valuation given by $v^{z_1}$ and $c_1$ plays $\top$ witnesses that $q_1 \models [A \cup C] \bigcirc \mathsf{sat}$ (and thus also $\mathfrak{M}, q_1 \models \chi_r$ for $r \ge 1$).

For the inductive step suppose the assumption holds up to index $i \ge 1$.

"$\Rightarrow$": Suppose $\mathfrak{M}, q_{i+1} \models \chi_{r+1}$ for $r \ge i$. Firstly, we prove the following claim.

***Claim:*** *Suppose* $\mathfrak{M}, q_{i+1} \models \chi_{r+1}$*, then each* $c_l$ *with* $l \le i$ *plays according to the valuation* $v(z_l)$*.*

*Proof.* Suppose $c_l$ plays $\perp$ and $d$ plays $l$. Then, the next state of the system is $\bar{q}_l$ and as a consequence, $\mathfrak{M}, q_l \models \neg\chi_r$ and by induction hypothesis $v(z_l) = \perp$.

The other case is proven by induction. Suppose $i = 1$, $\mathfrak{M}, q_2 \models \chi_{r+1}$, and $c_1$ plays $\top$. We have to show that $v(z_1) = \top$. Suppose the contrary. Then, for any strategy of $A \cup C$ there is a strategy of $B$ such that $\psi_1'$ evaluates false. Hence, if $d$ plays 1 rule (2) is firing and the next state is $q_\perp$ and thus $\mathfrak{M}, q_2 \not\models \chi_{r+1}$. Contradiction!

For the induction step, suppose that all agents $c_l$ for $l < i$ play according to $v(z_l)$, that $\mathfrak{M}, q_{i+1} \models \chi_{r+1}$, and $c_i = \top$. We show that $v(z_i) = \top$. For the sake of contradiction, suppose that $v(z_i) = \perp$. Again, for any strategy of $A \cup C$ witnessing $\chi_{r+1}$ we have that there is a strategy of $B$ that falsifies $\psi_i'$ (note, that by assumption $c_1, \ldots, c_{i-1}$ play according to $v(z_1), \ldots, v(z_{i-1})$). So, if $d$ plays $i$ rule (2) is firing and the next state is $q_\perp$ which implies $\mathfrak{M}, q_{i+1} \not\models \chi_{r+1}$. Contradiction! ■

Now let $s_{AC}$ be the strategy of agents $A \cup C$ that witnesses $\chi_{r+1}$ in $q_{i+1}$. Suppose player $d$ plays $i+1$. Irrelevant of the move of $c_{i+1}$ either rule (1) or rule (2) is firing. This does only depend on the valuation of $\psi_{i+1}'$. By assumption, we must have that $\psi_{i+1}'$ is true for all strategies of $B$ otherwise $\mathfrak{M}, q_{i+1} \not\models \chi_{r+1}$. Because of the previous claim, we must also have that $v(z_{i+1}) = \top$.

"$\Leftarrow$": Suppose $v(z_{i+1}) = \top$. Let $s_{AC}$ be the strategy in which players $c_j$ play according to $v(z_j)$ and players $a_j^o$ play according to $v^{z_j}$ if $v(z_j) = \top$ and

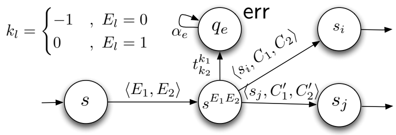



**Fig. B.4.** Transformation of transitions $(s, E_1, E_2)\Delta(s_i, C_1, C_2)$ and $(s, E_1, E_2)\Delta(s_j, C'_1, C'_2)$ .

arbitrarily if $v(z_j) = \bot$ for $o = 1, \ldots, s$. Suppose player $d$ plays $l \leq i+1$. Now, if each $c_j$ for $j = i, \ldots, l$ plays $\top$ we have that $\psi'_l$ is true as there is no valuation of variables $Y_l$ that makes $\psi_l$ false given the choices of $A \cup C$; hence, the next state is $q_\top$. Secondly, if $d$ plays $l$ and there is some agent $c_j$, $j > l$, that plays $\bot$; then rule (4) fires and the next state is also $q_\top$; the same holds if $d$ plays $l > i+1$. Finally, suppose $d$ plays $l$ and $c_l = \bot$. Then, by the definition of the actions of agents $C$, $v(z_l) = \bot$ and by induction hypothesis $\mathfrak{M}, q_l \models \neg\chi_r$; thus, $\mathfrak{M}, \bar{q}_l \models \mathsf{neg} \wedge [\emptyset]\neg\chi_r$ is true. Taking all theses cases together we have $\mathfrak{M}, q_{i+1} \models \chi_{r+1}$. ■

This gives us the following polynomial reduction:

$$z_m = \top \text{ iff } \mathfrak{M}_\mathcal{I}, q_m \models \chi_m.$$

■

## B.7 Model Checking **RAL***

### B.7.1 Non-Resource-Flat Languages

**Theorem 11.18 (⇝ page 275).** *Model checking* $\mathsf{RAL}_r$ *is undecidable, even in the single agent case; hence also,* $\mathsf{RAL}^+_r$ *and* $\mathsf{RAL}^*_r$ *are undecidable.*

*Proof.* Given a TCA $\mathcal{A} = (S, \Gamma, s^{\text{init}}, S_f, \Delta)$ we construct an RBAM $\mathfrak{M}^\mathcal{A}$ with two resources $R_1$ and $R_2$ (one per counter). We set $Q_{\mathfrak{M}^\mathcal{A}} = S \cup \{s^{E_1 E_2} \mid s \in S, E_1, E_2 \in \{0,1\}\} \cup \{q_e, q_a\}$. State $q_e$ (resp. $q_a$) is labelled $\mathsf{err}$ (resp. $\mathsf{halt}$) and represents the 'error' (resp. 'halting') state. The states $s^{E_1 E_2}$ are temporary states encoding that counter $k$ is zero ($E_k = 0$) or non-zero ($E_k = 1$) for $k = 1, 2$.

For each transition $(s, E_1, E_2)\Delta(s', C_1, C_2)$ of the automaton we introduce actions $\langle E_1, E_2 \rangle$ and $\langle s', C_1, C_2 \rangle$ (cf. Figure B.4). The first action leads from $s$ to $s^{E_1 E_2}$ and the second action from $s^{E_1 E_2}$ to $s'$. Action $\langle s', C_1, C_2 \rangle$ consumes/produces $C_i$ units of resource $R_i$, $i = 1, 2$. The other kinds of actions are cost-free. Clearly, actions can only be performed if sufficient resources are available. We need to ensure that actions $\langle E_1, E_2 \rangle$ with some $E_i = 0$ can only be performed if the counter $i$ is actually 0; that is, if *no* resources of type $R_i$ are available. Therefore, special 'test' actions $t_{k_2}^{k_1}$ that cost $k_i$ units of resource $R_i$ are introduced, $k_i \in \{0, -1, 1\}$. Such actions can only be performed in states $s^{E_1E_2}$ with some $E_i = 0$ and they always lead to state $q_e$. Now, in a state $s^{E_1E_2}$ with some element equal 0, say $E_1 = 0$, $E_2 = 1$, (representing that counter 1 should be zero and 2 be non-zero) action $t_0^{-1}$ can be used to verify whether the currently available resources model the counter correctly: If $q_e$ is reachable resources of type $R_1$ are available although this should not be the case according to $E_1$. Moreover, we add an action $\alpha_e$ to state $q_e$, leading back to $q_e$ and an action $\alpha_a$ that leads from any state $s \in S_f$ to $q_a$ and from $q_a$ to itself. We assume that these are the only actions in states $q_e$ and $q_a$ and that they will be executed by default.

We show: $\mathcal{A}\downarrow$ iff $\mathfrak{M}^{\mathcal{A}}, s^{\text{init}}, \eta_0 \models_r \neg\langle\langle\emptyset\rangle\rangle_{\mathbb{A}\text{gt}}^{\eta_0}\neg((\neg\langle\langle\emptyset\rangle\rangle \bigcirc \neg\mathsf{err})\mathcal{U}\mathsf{halt})$.

The formula states that there is an $(\eta_0, s_\emptyset, \mathbb{A}\text{gt})$-path such that eventually halt and the error state can never be reached along the way to $q_a$.

"$\Rightarrow$": Let $\delta = (s_i, v_1^i, v_2^i)_{i=1,\dots k}$ be an accepting configuration. Clearly, if agent 1 executes $\langle E_1^i, E_2^i\rangle$ in state $s_i \notin S_f$, action $\langle s_{i+1}, C_1^i, C_2^i\rangle$ in state $s_i^{E_1^iE_2^i}$ (according to $\delta_i$ as introduced above), and $\alpha_a$ in $s_k \in S_f$ the resulting path is given by $\lambda$ with $\lambda|_Q = (s_j s_j^{E_1^jE_2^j} s_{j+1})_{j=1,\dots,k-1}(q_a)^\omega$. It remains to show that for any state $s_i^{E_1^iE_2^i}$ with $E_l^i = 0$ we have that $\lambda|_{\mathcal{R}es}[2i-1](1, R_l) = 0$ (i.e. in this state agent 1 has no resources of type $R_l$). By induction one can easily prove that the actions keep track of the resources correctly and thus action $t_0^{-1}$ cannot be executed in any $s_j^{E_1^jE_2^j}$ along the path.

**Claim:** For each $3j < k$ with $j \geq 0$ and $\lambda[3j] = (s_{j+1}, \eta_{j+1})$ we have that $\eta_{j+1}(1, R_i) = v_i^{j+1}$ for $i = 1, 2$.

*Proof.* [of Claim] Proof by induction. Clearly, $\eta_0(1, R_i) = \eta_1(1, R_i) = v_i^1 = 0$, for $i = 1, 2$. Suppose the claim is correct for $3(j-1)+1$. Then, agent 1 can perform action $(s_j, C_1^jC_2^j)$ in $s_j^{E_1^jE_2^j}$. This action costs $C_i^j$ of resource $R_i$ for $i = 1, 2$. In the automaton the transition $\delta_j = ((s_j, E_1^jE_2^j), (s_{j+1}, C_1^j, C_2^j))$ is taken. Hence, we have $\eta_{j+1}(1, R_i) = \eta_j(1, R_i) + C_i^j = v_i^j + C_i^j = v_i^{j+1}$ for $i = 1, 2$. ■

"$\Leftarrow$": Clearly, if such a satisfying path exists it must have the structure as shown above and we can directly construct an accepting computation of the automaton. Each triple $s_i s_i^{E_1^iE_2^i} s_{i+1}$ uniquely determines a transition $\delta_i$. That only valid transitions are chosen is shown in the same way as for the left-to-right direction. ■

**Theorem 11.19 (****$\rightsquigarrow$ page 275****).** *Model checking pr-*$\mathbf{RAL}_R$ *(even without the release operator) is undecidable in the single-agent case; hence, also pr-*$\mathbf{RAL}_R^+$*, pr-*$\mathbf{RAL}_R^*$*,*$\mathbf{RAL}_R$*,* $\mathbf{RAL}_R^+$*, and* $\mathbf{RAL}_R^*$ *are undecidable.*

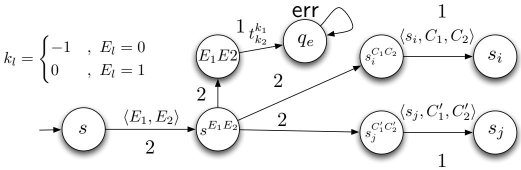



**Fig. B.5.** Construction used in the proof of Theorem 11.20 for $(s, E_1, E_2)\Delta(s_i, C_1, C_2)$ and $(s, E_1, E_2)\Delta(s_j, C'_1, C'_2)$.

*Proof.* We use the very same construction and notation as in the proof of Theorem 11.18. We show $\mathcal{A}\downarrow$ iff $\mathfrak{M}^{\mathcal{A}}, s^{\text{init}}, \eta_0 \models_R \langle\langle 1\rangle\rangle^{\eta_0}((\neg\langle\langle 1\rangle\rangle \bigcirc \mathsf{err})\mathcal{U}\mathsf{halt})$.
"⇒": Agent 1's strategy is given by the same strategy as constructed before. That it correctly keeps track of the resources is also shown analogously. Now, in each state $s_i^{E_1^iE_2^i}$ the agent tries to execute an appropriate test action to reach the error state. However, this action will never be activated in a valid computation of the automaton.
"⇐" Such a winning strategy of 1 does also directly imply an accepting computation of the automaton. ■

**Theorem 11.20 (⇝ page 275).** *Model checking pr-$\mathbf{RAL}_r$ is undecidable for models with at least two agents; hence, also pr-$\mathbf{RAL}^+_r$ and pr-$\mathbf{RAL}^*_r$ are undecidable.*

*Proof.* The model $\mathfrak{M}^{\mathcal{A}}$ considered here is more sophisticated than the ones before and shown in Figure B.5. The error state $q_e$ is not reachable directly from the test state $s^{E_1E_2}$; we rather add an intermediate state $E_1E_2$ with $E_i \in \{0, 1\}$; that is, the model contains 4 additional states. From these new states the error state is reached. The new model is turn-based; transitions are labelled with the agent who can make the choices. In each state in which it is agent 1's turn, there is only a single action available. The actions have the same costs as before. Agent 2's actions are cost-free. The idea is that agent 2 makes the choices. In the resulting state in which it is agent 1's turn agent 1 does only has a unique choice which is used to keep track of the resources. Due to the unique choice a memoryless strategy for 1 suffices.

We show $\mathcal{A}\downarrow$ iff $\mathfrak{M}^{\mathcal{A}}, s^{\text{init}}, \eta_0 \models_r \neg\langle\langle 1\rangle\rangle^{\eta_0}\neg((\neg\langle\langle 2\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \bigcirc \mathsf{err})\mathcal{U}\mathsf{halt})$. We have to show that for all strategies of 1 (where 1 never has a choice which allows to use $r$-strategies) there is a path $\lambda$ (that completely depends on agent 2) such that state $q_a$ is reached and on the way to this state whenever 2 could decide to enter the new state $(E_1E_2)$ agent 1 has not enough resources to enter the error state.
"⇒": Let $\delta = (s_i, v_1^i, v_2^i)_{i=1,\ldots k}$ be an accepting configuration. Then, in each state $s_i \notin S_f$ corresponding to a state of the automaton, agent 2 performs

action $\langle E_1^i, E_2^i\rangle$ and in state $s_i^{E_1^i,E_2^i}$ the action leading to $s_{i+1}^{C_1^iC_2^i}$. This results in path $\lambda$ with:

$$\lambda|_Q = (s_i s_i^{E_1^iE_2^i} s_{i+1}^{C_1^iC_2^i} s_{i+1})_{i=1,\dots,k-1}(q_a)^\omega. \qquad (\star)$$

Analogously to the claim in the proof of Theorem B.5 we get the following result.

**Claim:** For each $4j < k$, $j \geq 0$ with $\lambda[4j] = (s_{j+1}, \eta_{j+1})$ we have that $\eta_j(1, R_i) = v_i^{j+1}$ for $i = 1, 2$.

*Proof.* [of Claim] Proof by induction. Clearly, $\eta_0(1, R_i) = \eta_1(1, R_i) = v_i^1 = 0$, for $i = 1, 2$. Suppose the claim is correct for $4(j-1)$ and let $\lambda[4j] = (s_{j+1}, \eta_{j+1})$ be the next state. Then, agent 1 has performed action $(s_{j+1}, C_1^jC_2^j)$ in $s_{j+1}^{C_1^jC_2^j}$. This action costs $C_i^j$ of resource $R_i$ for $i = 1, 2$. In the automaton the transition $\delta_j = ((s_j, E_1^jE_2^j), (s_{j+1}, C_1^j, C_2^j))$ is taken. Hence, we have $\eta_{j+1}(1, R_i) = \eta_j(1, R_i) + C_i^j = \eta_j(1, R_i) + C_i^j = v_i^j + C_i^j = v_i^{j+1}$ for $i = 1, 2.$ ■

According to the claim, we have that $\neg\langle\langle 2\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \bigcirc \mathsf{err}$ whenever a state $s_i^{E_1^iE_2^i}$ is visited along the path with $E_j = 0$ for some $j \in \{1, 2\}$.

"$\Leftarrow$": Suppose the formula holds. Then, the path leading to state $q_a$ must have the structure given in ($\star$). Again, we can identify each quadruple of states $s_i s_i^{E_1^iE_2^i} s_{i+1}^{C_1^iC_2^i} s_{i+1}$ with a transition $\delta_i$. Analogously to the claim proven above, we obtain an accepting configuration of $\mathcal{A}$ in which each transition is chosen correctly. ■

### B.7.2 Undecidability: Resource-Flat Languages

**Theorem 11.21 (⇝ page 276).** *Model checking rf-*$\mathsf{RAL}_R$ *is undecidable for models with at least two agents; thus, also rf-*$\mathsf{RAL}_R^+$ *and rf-*$\mathsf{RAL}_r^*$ *are undecidable.*

*Proof.* The idea is similar to the proof of Theorem 11.19 but the test whether the error state is reachable is performed by the opponent (agent 2). That is, we add a second agent to the model shown in Figure B.4 that can execute the test action in states $s^{E_1E_2}$. The task of agent 1 is to prevent agent 2 to perform such actions. Analogously to the proof of Theorem 11.19 we show that the transitions in the model correctly keep track of resources. We have: $\mathcal{A}\downarrow$ iff $\mathfrak{M}^{\mathcal{A}}, s^{\text{init}}, \eta_0 \models_R \langle\langle 1\rangle\rangle^{\eta_0}_{\mathbb{A}\text{gt}} \Diamond \mathsf{halt}$. The only way to avoid $\mathsf{halt}$ is if there is path corresponding to a run of the automaton that does not halt or if the opponent can move to state $q_e$, the latter can be prevented by 1 choosing the right transitions. ■

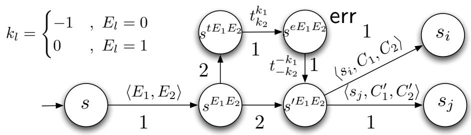



**Fig. B.6.** Construction used in the proof of Theorem 11.22 for $(s, E_1, E_2)\Delta(s_i, C_1, C_2)$ and $(s, E_1, E_2)\Delta(s_j, C_1', C_2')$.

**Theorem 11.22** (⇝ page 276). *Model Checking rf-pr-*$\mathbf{RAL}^*_R$*, rf-pr-*$(\mathcal{L}_{RAL}, \models^\infty_R)$*, and rf-pr-*$(\mathcal{L}_{RAL}, \models^\infty_R)$ *is undecidable for models with at least two agents.*

*Proof.* For the reduction we construct a model as shown in Figure B.6; we will avoid formal details and just sketch the main idea. Analogously to the proof of Theorem 11.21 we use agent 1 to simulate the transitions in the automaton. As before, in the test states $s^{E_1E_2}$ agent 2 tries to falsify the computation by entering the "test loop"; but, because of proponent-restrictiveness agent 2 may always be successful with this very action. Hence, if agent 2 performs the test action $t^{k_1}_{k_2}$ and the intermediate state $s^{tE_1E_2}$ is reached in which it is up to agent one to reach the error state. Since agent 1 does only have a single action available it has to take it if enough resources are available due to the maximality conditions on paths. Once the error state is reached agent 1 has to perform an action which adds the consumed resources of the test action (it can be seen as the reverse function).

It is important to note, that if agent 2 performs the test action and there are not sufficient resource of 1 to enter the error state the path is deemed to be finite and thus is disregarded from the outcome. Hence, the error state labelled err does only occur in the outcome if it is part of an infinite path which in turn can only happen if agent 1 has no strategy that corresponds to an accepting configuration of the automaton. ∎

# C

# Curriculum Vitae

## Personal Data

| | |
|---|---|
| Degree/Name | Dr. rer. nat. Nils Bulling |
| Birth | April 2nd, 1981 in Hamburg, Germany |
| Nationality | German |

## Education / Degrees

| | |
|---|---|
| 2001–2006 | *Study of Computer Science*, TU Clausthal, Germany<br>Vordiplom (finishing undergraduate) dated October 7, 2003.<br>Diploma in Computer Science dated March 20, 2006<br>(Mark: *1.0 (with honours)*)<br>Diploma-Thesis: *Modal Logics for Games, Time, and Beliefs* |
| 2003–2004 | *Study of Mathematics*, TU Clausthal, Germany<br>Vordiplom (finishing undergraduate) dated December 4, 2004<br>(Mark: *1.2*). |
| 2005–2006 | *Study of Computer Science*, Durham University, UK |
| 2006–2010 | *PhD Studies in Computer Science*, TU Clausthal, Germany<br>Doctoral examination (Dr. rer. nat.) dated October 22, 2010<br>(Mark: *summa cum laude (with honours)*) .<br>PhD-Committee: Prof. Dr. Jürgen Dix (promoter), PD Dr. habil. Wojciech Jamroga (co-promoter), Prof. Dr. Sarit Kraus (co-promoter), Prof. Dr. Jörg Müller (chair).<br>Title of PhD-Thesis: *Modelling and Verifying Abilities of Rational Agents.* |

## Employment History / Professional Experience

| | |
|---|---|
| 2002–2005 | *Student Assistant*, TU Clausthal, Germany |
| 2004–2005 | *Teaching Assistant*, TU Clausthal, Germany |
| 2006–2010 | *Research Assistant* (full-time employed), TU Clausthal, Germany |
| 2010–present | *Postdoc/Research Associate*, TU Clausthal, Germany |

| | |
|---|---|
| 2008–present | *Chairman of the Scientific Employees*, TU Clausthal, Germany |
| 2008–present | *Deputy Member of the Faculty of Computer Science, Mathematics and Engineering*, TU Clausthal, Germany |
| 2009–present | *Assistant of the Executive Board of the Department of Informatics*, TU Clausthal, Germany |

# References


Ågotnes, T., Goranko, V., and Jamroga, W. (2007a). Alternating-time temporal logics with irrevocable strategies. In Samet, D., editor, *Proceedings of TARK XI*, pages 15–24.

Ågotnes, T., van der Hoek, W., and Wooldridge, M. (2007b). Quantified coalition logic. In *IJCAI*, pages 1181–1186.

Ågotnes, T. and Walther, D. (2009). A logic of strategic ability under bounded memory. *J. of Logic, Lang. and Inf.*, 18(1):55–77.

Alechina, N., Logan, B., and Hoang, N. (2009a). Verifying properties of coalitional ability under resource bounds. In Farwer, B., editor, *Proceedings of the Second Internatinoal Workshop on Logics for Agents and Mobility (LAM'09)*, Los Angeles CA, USA.

Alechina, N., Logan, B., Nga, N. H., and Rakib, A. (2008). Verifying time, memory and communication bounds in systems of reasoning agents. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 736–743. International Foundation for Autonomous Agents and Multiagent Systems.

Alechina, N., Logan, B., Nga, N. H., and Rakib, A. (2009b). A logic for coalitions with bounded resources. In Boutilier, C., editor, *Proceedings of the Twenty First International Joint Conference on Artificial Intelligence*, volume 2, pages 659–664, Pasadena CA, USA. IJCAI/AAAI, AAAI Press.

Alechina, N., Logan, B., Nga, N. H., and Rakib, A. (2010). Resource-bounded alternating-time temporal logic. In van der Hoek, W., Kaminka, G., Lespérance, Y., Luck, M., and Sen, S., editors, *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, Toronto, Canada. IFAAMAS. (to appear).

Alur, R., Henzinger, T., Mang, F., Qadeer, S., Rajamani, S., and Tasiran, S. (1998a). MOCHA user manual. In *Proceedings of CAV'98*, volume 1427 of *Lecture Notes in Computer Science*, pages 521–525.

Alur, R., Henzinger, T. A., and Kupferman, O. (1997). Alternating-time Temporal Logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 100–109. IEEE Computer Society Press.

Alur, R., Henzinger, T. A., and Kupferman, O. (1998b). Alternating-time Temporal Logic. *Lecture Notes in Computer Science*, 1536:23–60.

Alur, R., Henzinger, T. A., and Kupferman, O. (2002). Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713.

Amgoud, L. (2005a). An argumentation-based model for reasoning about coalition structures. In *ArgMAS*, pages 217–228.

Amgoud, L. (2005b). Towards a formal model for task allocation via coalition formation. In *AAMAS*, pages 1185–1186.

Angluin, D. (1980). On counting problems and the polynomial-time hierarchy. *Theor. Comput. Sci.*, 12:161–173.

Bacchus, F. and Kabanza, F. (1998). Planning for temporally extended goals. *Ann. Math. Artif. Intell.*, 22(1-2):5–27.

Bacchus, F. and Kabanza, F. (2000). Using temporal logics to express search control knowledge for planning. *Artificial Intelligence*, 116(1-2):123–191.

Bacharach, M. (1987). A theory of rational decision in games. *Erkenntnis*, 27:17–55.

Baier, C. and Katoen, J.-P. (2008). *Principles of Model Checking*. The MIT Press.

Baltag, A. (2002). A logic for suspicious players. *Bulletin of Economic Research*, 54(1):1–46.

Baral, C., Kreinovich, V., Sarkar, S., and Trejo, R. (2001). Computational complexity of planning with temporal goals. In *IJCAI*, pages 509–514.

Baral, C. and Zhao, J. (2007). Non-monotonic temporal logics for goal specification. In *IJCAI*, pages 236–242.

Beeri, C. (1980). On the menbership problem for functional and multivalued dependencies in relational databases. *ACM Trans. Database Syst.*, 5(3):241–259.

Beigel, R., Reingold, N., and Spielman, D. (1995). Pp is closed under intersection. *Journal of Computer and System Sciences*, 50:1–9.

Bellman, R. (1957). A Markovian decision process. *Journal of Mathematics and Mechanics*, 6:679–684.

Blackburn, P., Benthem, J. F. A. K. v., and Wolter, F. (2006). *Handbook of Modal Logic, Volume 3 (Studies in Logic and Practical Reasoning)*. Elsevier Science Inc., New York, NY, USA.

Bonanno, G. (1991). The logic of rational play in games of perfect information. *Economics and Philosophy*, 7:37–65.

Bonanno, G. (2002). Modal logic and game theory: Two alternative approaches. *Risk Decision and Policy*, 7:309–324.

Borgo, S. (2007). Coalitions in action logic. In *IJCAI*, pages 1822–1827.

Bratman, M. (1987). *Intentions, Plans, and Practical Reason.* Harvard University Press.

Brena, R. F., Aguirre, J.-L., Chesñevar, C. I., Ramírez, E. H., and Garrido, L. (2007). Knowledge and information distribution leveraged by intelligent agents. *Knowl. Inf. Syst.*, 12(2):203–227.

Brihaye, T., Costa, A. D., Laroussinie, F., and Markey, N. (2008). ATL with strategy contexts and bounded memory. Technical Report LSV-08-14, ENS Cachan.

Broersen, J., Herzig, A., and Troquard, N. (2006). A stit-extension of atl. In *JELIA*, pages 69–81.

Bulling, N. (2010). Model checking coalition logic on implicit models is $\Delta_3$-complete. Technical report, Clausthal University of Technology.

Bulling, N., Chesnevar, C., and Dix, J. (2008). Modelling Coalitions: ATL + Argumentation. In Padgham, L. and Parkes, D., editors, *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 681–688, Estoril, Portugal. ACM Press.

Bulling, N. and Dix, J. (2008). A finer grained modeling of rational coalitions using goals. In Norberto Caminoa, Fernanda Carmona, A. C. L., editor, *Proceedings of the 14th Argentinean Conference on Computer Science CACIC'08*, Chilecito, La Rioja, Argentina. Red UNCI.

Bulling, N. and Dix, J. (2010). Modelling and verifying coalitions using argumentation and ATL. *Inteligencia Artificial*, 14(46):45–73.

Bulling, N., Dix, J., and Chesñevar, C. (2009a). An argumentative approach for modelling coalitions using ATL. In Rahwan, I. and Moraitis, P., editors, *Proc. of the 5rd. Intl. Workshop on Argumentation and Multiagent Systems ArgMAS 2008 (Selected Contributions and Invited Papers)*, Lecture Notes, pages 190–211, Estoril, Portugal. Springer.

Bulling, N., Dix, J., and Jamroga, W. (2010). Model checking logics of strategic ability: Complexity. In Dastani, M., Hindriks, K. V., and Meyer, J.-J. C., editors, *Specification and Verification of Multi-Agent Systems.* Springer.

Bulling, N. and Farwer, B. (2010a). Expressing properties of resource-bounded systems: The logics RBTL and RBTL*. In Dix, J., Fisher, M., and Novak, P., editors, *Post-Proceedings of CLIMA '09*, Hamburg, Germany.

Bulling, N. and Farwer, B. (2010b). On the decidability of verifying resource-bounded agents. Technical Report IfI-10-05, Clausthal University of Technology.

Bulling, N. and Farwer, B. (2010c). On the (Un-)Decidability of Model-Checking Resource-Bounded Agents. In Coelho, H. and Wooldridge, M., editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, pages 567–572, Porto, Portugal.

Bulling, N. and Jamroga, W. (2006). Agents, beliefs, and plausible behaviour in a temporal setting. Technical Report IfI-06-05, Clausthal University of Technology.

Bulling, N. and Jamroga, W. (2007a). Agents, beliefs, and plausible behavior in a temporal setting. In *Proceedings of AAMAS'07*, pages 570–577, Honolulu, Hawaii, USA. ACM Press.

Bulling, N. and Jamroga, W. (2007b). A logic for reasoning about rational agents: Yet another attempt. In Czaja, L., editor, *Proceedings of the International Workshop on Concurrency, Specification and Programming CS&P'07*, pages 87–99, Lagow, Poland.

Bulling, N. and Jamroga, W. (2008). What agents can probably enforce. In *Proceedings of the International Workshop on Concurrency, Specification and Programming CS&P'08*, Gross Vaeter, Germany.

Bulling, N. and Jamroga, W. (2009a). Model checking $\mathrm{ATL}^+$ is harder than it seemed. In *Proceedings of the 7th European Workshop on Multi-Agent Systems EUMAS'09*, Ayia Napa, Cyprus.

Bulling, N. and Jamroga, W. (2009b). Rational play and rational beliefs under uncertainty. In *Proceedings of AAMAS'09*, pages 257–264, Budapest, Hungary. ACM Press.

Bulling, N. and Jamroga, W. (2009c). What agents can probably enforce. *Fundamenta Informaticae*, 93:81–96.

Bulling, N. and Jamroga, W. (2010a). Verifying agents with memory is harder than it seemed. In *Proceedings of AAMAS 2010*, pages 699–706, Toronto, Canada. ACM Press.

Bulling, N. and Jamroga, W. (2010b). Verifying agents with memory is harder than it seemed. *AI-Communications*, (to appear).

Bulling, N., Jamroga, W., and Dix, J. (2009b). Reasoning about temporal properties of rational play. *Annals of Mathematics and Artificial Intelligence*, 53(1-4):51–114.

Caminada, M. (2006). Semi-stable semantics. In *Intl. Conference on Computational Models of Argument (COMMA)*, pages 121–130.

Chesñevar, C. I., Maguitman, A. G., and Loui, R. P. (2000). Logical models of argument. *ACM Comput. Surv.*, 32(4):337–383.

Chu, F. and Halpern, J. (2001). On the NP-completeness of finding an optimal strategy in games with common payoffs. *International Journal of Game Theory*.

Clarke, E. and Emerson, E. (1981). Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proceedings of Logics of Programs Workshop*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71.

Clarke, E., Emerson, E., and Sistla, A. (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263.

Clarke, E., Grumberg, O., and Peled, D. (1999). *Model Checking*. MIT Press.

Conitzer, V. and Sandholm, T. (2002). Complexity results about Nash equilibria. Technical Report CMU-CS-02-135, School of Computer Science, Carnegie-Mellon University.

Dastani, M. and Jamroga, W. (2010). Reasoning about strategies of multi-agent programs. In *Proceedings of AAMAS2010*, pages 997–1004.

de Alfaro, L., Faella, M., Henzinger, T., Majumdar, R., and Stoelinga, M. (2004). Model checking discounted temporal properties. In *Proceedings of TACAS'04*, volume 2988 of *LNCS*, pages 57–68.

Dembiński, P., Janowska, A., Janowski, P., Penczek, W., Półrola, A., Szreter, M., Woźna, B., and Zbrzezny, A. (2003). Verics: A tool for verifying timed automata and estelle specifications. In *Proceedings of the of the 9th Int. Conf. on Tools and Algorithms for Construction and Analysis of Systems (TACAS'03)*, volume 2619 of *LNCS*, pages 278–283. Springer.

Dimopoulos, Y. and Torres, A. (1996). Graph theoretical structures in logic programs and default theories. *Theor. Comput. Sci.*, 170(1-2):209–244.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358.

Dunne, P. E. and Caminada, M. (2008). Computational complexity of semi-stable semantics in abstract argumentation frameworks. In *JELIA '08: Proceedings of the 11th European conference on Logics in Artificial Intelligence*, pages 153–165, Berlin, Heidelberg. Springer-Verlag.

Emerson, E. and Halpern, J. (1982). Sometimes and not never revisited: On branching versus linear time temporal logic. In *Proceedings of the Annual ACM Symposium on Principles of Programming Languages*, pages 151–178.

Emerson, E. and Halpern, J. (1985). Decision procedures and expressivenessin the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1–24.

Emerson, E. and Halpern, J. (1986). Sometimes and not never revisited: On branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178.

Emerson, E. A. (1990). Temporal and modal logic. In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier Science Publishers.

Emerson, E. A. and Jutla, C. S. (1988). The complexity of tree automata and logics of programs. In *SFCS '88: Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 328–337, Washington, DC, USA. IEEE Computer Society.

Emerson, E. A. and Lei, C.-L. (1987). Modalities for model checking: Branching time logic strikes back. *Science of Computer Programming*, 8(3):275–306.

Emerson, E. A. and Sistla, A. P. (1984). Deciding branching time logic. In *STOC '84: Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 14–24, New York, NY, USA. ACM.

Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y. (1995). *Reasoning about Knowledge*. MIT Press: Cambridge, MA.

Fischer, M. J. and Ladner, R. E. (1979). Propositional dynamic logic of regular programs. *J. Comput. Syst. Sci.*, 18(2):194–211.

Friedman, N. and Halpern, J. (1994). A knowledge-based framework for belief change, Part I: Foundations. In *Proceedings of TARK*, pages 44–64.

Furst, M., Saxe, J. B., and Sipser, M. (1984). Parity, circuits, and the polynomial-time hierarchy. *Math. Systems Theory*, 17:13–ñ27.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman: San Francisco.

Gilboa, I. and Zemel, E. (1989). Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*.

Gill, J. (1977). Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4).

Goranko, V. and Jamroga, W. (2004). Comparing semantics of logics for multi-agent systems. *Synthese*, 139(2):241–280.

Goranko, V. and van Drimmelen, G. (2003). Complete axiomatization and decidability of the Alternating-time Temporal Logic. Submitted.

Harding, A., Ryan, M., and Schobbens, P.-Y. (2002). Approximating ATL* in ATL. In *VMCAI '02: Revised Papers from the Third International Workshop on Verification, Model Checking, and Abstract Interpretation*, pages 289–301, London, UK. Springer-Verlag.

Harrenstein, B., van der Hoek, W., Meyer, J., and Witteveen, C. (2003). A modal characterization of Nash equilibrium. *Fundamenta Informaticae*, 57(2–4):281–321.

Harrenstein, P., van der Hoek, W., Meijer, J., and Witteveen, C. (2002). Subgame-perfect Nash equilibria in dynamic logic. In Pauly, M. and Baltag, A., editors, *Proceedings of the ILLC Workshop on Logic and Games*, pages 29–30. University of Amsterdam. Tech. Report PP-1999-25.

Hattori, H., Ito, T., Ozono, T., and Shintani, T. (2001). An approach to coalition formation using argumentation-based negotiation in multi-agent systems. In *Proceedings of IEA/AIE*, pages 687–696, London, UK. Springer-Verlag.

Herzig, A. and Troquard, N. (2006). Knowing how to play: Uniform choices in logics of agency. In *Proceedings of AAMAS'06*.

Hintikka, J. (1973). *Logic, Language Games and Information*. Clarendon Press : Oxford.

Hintikka, J. and Sandu, G. (1997). Game-theoretical semantics. In van Benthem, J. and ter Meulen, A., editors, *Handbook of Logic and Language*, pages 361–410. Elsevier: Amsterdam.

Hopcroft, J. and Ullman, J. (1979). *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, Massachusetts.

Immerman, N. (1981). Number of quantifiers is better than number of tape cells. *Journal of Computer and System Sciences*, 22(3):384 – 406.

Jamroga, W. (2003). Some remarks on alternating temporal epistemic logic. In Dunin-Keplicz, B. and Verbrugge, R., editors, *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, pages 133–140.

Jamroga, W. (2004). *Using Multiple Models of Reality. On Agents who Know how to Play Safer.* PhD thesis, University of Twente.

Jamroga, W. (2007). Reducing knowledge operators in the context of model checking. Technical Report IfI-07-09, Clausthal University of Technology.

Jamroga, W. (2008a). A temporal logic for multi-agent MDP's. In *Proceedings of the AAMAS Workshop on Formal Models and Methods for Multi-Robot Systems*, pages 29–34.

Jamroga, W. (2008b). A temporal logic for stochastic multi-agent systems. In *Proceedings of PRIMA'08*, volume 5357 of *LNCS*, pages 239–250.

Jamroga, W. and Ågotnes, T. (2006). What agents can achieve under incomplete information. In *Proceedings of AAMAS'06.* To appear.

Jamroga, W. and Ågotnes, T. (2007). Constructive knowledge: What agents can achieve under incomplete information. *Journal of Applied Non-Classical Logics*, 17(4):423–475.

Jamroga, W. and Bulling, N. (2007a). A general framework for reasoning about rational agents. In *Proceedings of AAMAS'07*, pages 592–594, Honolulu, Hawaii, USA. ACM Press.

Jamroga, W. and Bulling, N. (2007b). A logic for reasoning about rational agents. In Sadri, F. and Satoh, K., editors, *Computational Logic in Multi-Agent Systems, 8th International Workshop, CLIMA VIII, Revised Selected and Invited Papers*, volume 5056 of *Lecture Notes*, pages 42–61, Porto, Portugal. Springer.

Jamroga, W. and Dix, J. (2005). Do agents make model checking explode (computationally)? In Pěchouček, M., Petta, P., and Varga, L., editors, *Proceedings of CEEMAS 2005*, volume 3690 of *Lecture Notes in Computer Science*, pages 398–407. Springer Verlag.

Jamroga, W. and Dix, J. (2006). Model checking $\mathrm{ATL}_{ir}$ is indeed $\Delta_2^P$-complete. In *Proceedings of EUMAS'06.*

Jamroga, W. and Dix, J. (2008). Model checking abilities of agents: A closer look. *Theory of Computing Systems*, 42(3):366–410.

Jamroga, W. and van der Hoek, W. (2004). Agents that know how to play. *Fundamenta Informaticae*, 63(2–3):185–219.

Jamroga, W., van der Hoek, W., and Wooldridge, M. (2005). Intentions and strategies in game-like scenarios. In Bento, C., Cardoso, A., and Dias, G., editors, *Progress in Artificial Intelligence: Proceedings of EPIA 2005*, volume 3808 of *Lecture Notes in Artificial Intelligence*, pages 512–523. Springer Verlag.

Jančar, P. (1990). Decidability of a temporal logic problem for petri nets. *Theor. Comput. Sci.*, 74(1):71–93.

Jones, N. D. (1975). Space-bounded reducibility among combinatorial problems. *Journal of Computer and System Sciences*, 11(1):68 – 85.

Jones, N. D. (1977). Corrigendum: Space-bounded reducibility among combinatorial problems. *J. Comput. Syst. Sci.*, 15(2):241.

Jonker, G. (2003). Feasible strategies in Alternating-time Temporal Epistemic Logic. Master thesis, University of Utrecht.

Kaci, S. and van der Torre, L. (2008). Preference-based argumentation: Arguments supporting multiple values. *Int. J. Approx. Reasoning*, 48(3):730–751.

Kaci, S., van der Torre, L. W. N., and Weydert, E. (2007). On the acceptability of incompatible arguments. In *Proc. of ECSQARU Conf.*, pages 247–258.

Karp, R. M. and Miller, R. E. (1969). Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195.

Karunatillake, N. C., Jennings, N. R., Rahwan, I., and Ramchurn, S. D. (2006). Managing social influences through argumentation-based negotiation. In *Proc. of AAMAS*, pages 426–428.

Kemeny, J. G., Snell, L. J., and Knapp, A. W. (1966). *Denumerable Markov Chains*. Van Nostrand.

Koller, D. and Megiddo, N. (1992). The complexity of twoperson zero-sum games in extensive form. *Games and Economic Behavior*, 4:528–552.

Kucera, A. and Strejcek, J. (2002). The stuttering principle revisited: On the expressiveness of nested x and u operators in the logic ltl. In *CSL '02: Proceedings of the 16th International Workshop and 11th Annual Conference of the EACSL on Computer Science Logic*, pages 276–291, London, UK. Springer-Verlag.

Kupferman, O., Vardi, M., and Wolper, P. (2000). An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360.

Laroussinie, F. (1995). About the expressive power of CTL combinators. *Information Processing Letters*, 54(6):343–345.

Laroussinie, F., Markey, N., and Oreiby, G. (2006). Expressiveness and complexity of ATL. Technical Report LSV-06-03, CNRS & ENS Cachan, France.

Laroussinie, F., Markey, N., and Oreiby, G. (2008). On the expressiveness and complexity of atl. *CoRR*, abs/0804.2435.

Laroussinie, F., Markey, N., and Schnoebelen, P. (2001). Model checking CTL+ and FCTL is hard. In *Proceedings of FoSSaCS'01*, volume 2030 of *Lecture Notes in Computer Science*, pages 318–331. Springer.

Lichtenstein, O. and Pnueli, A. (1985). Checking that finite state concurrent programs satisfy their linear specification. In *POPL '85: Proceedings of the*

*12th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 97–107, New York, NY, USA. ACM.

Lomuscio, A. and Sergot, M. (2003). Deontic interpreted systems. *Studia Logica*, 75(1):63–92.

Lomuscio, A. and Sergot, M. (2004). A formalisation of violation, error recovery, and enforcement in the bit transmission problem. *Journal of Applied Logic*, 2(1):93–116.

Mas-Colell, A., Whinston, M. D., and Green, J. R. (1995). *Microeconomic Theory*. Oxford.

Meyer, A. R. and Stockmeyer, L. J. (1972). The equivalence problem for regular expressions with squaring requires exponential space. In *SWAT '72: Proceedings of the 13th Annual Symposium on Switching and Automata Theory (swat 1972)*, pages 125–129, Washington, DC, USA. IEEE Computer Society.

Moses, Y. and Tennenholz, M. (1995). Artificial social systems. *Computers and AI*, 14(6):533–562.

Myers, G. J. and Sandler, C. (2004). *The Art of Software Testing*. John Wiley & Sons.

Osborne, M. and Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press.

Papadimitriou, C. (1994). *Computational Complexity*. Addison Wesley : Reading.

Parikh, R. (1985). The logic of games and its applications. *Ann. of Discrete Mathematics*, 24:111–140.

Pauly, M. (2002). A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166.

Penczek, W. and Lomuscio, A. (2003). Verifying epistemic properties of multi-agent systems via bounded model checking. In *Proceedings of AAMAS'03*, pages 209–216, New York, NY, USA. ACM Press.

Pnueli, A. (1977). The temporal logic of programs. In *Proceedings of FOCS*, pages 46–57.

Pnueli, A. and Rosner, R. (1989). On the synthesis of a reactive module. In *POPL '89: Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 179–190, New York, NY, USA. ACM.

Prakken, H. and Vreeswijk, G. (2002). Logical Systems for Defeasible Argumentation. In Gabbay, D. and F.Guenther, editors, *Handbook of Phil. Logic*, pages 219–318. Kluwer.

Queille, J.-P. and Sifakis, J. (1982). Specification and verification of concurrent systems in cesar. In *Proceedings of the 5th Colloquium on International Symposium on Programming*, pages 337–351, London, UK. Springer-Verlag.

Rabin, M. (1970). Weakly definable relations and special automata. *Mathematical Logic and Foundations of Set Theory*, pages 1–23.

Rahwan, I. and Amgoud, L. (2006). An argumentation based approach for practical reasoning. In *Proc. of AAMAS Conf.*, pages 347–354.

Rahwan, I., Pasquier, P., Sonenberg, L., and Dignum, F. (2007). On the benefits of exploiting underlying goals in argument-based negotiation. In *Proc. of AAAI Conf.*, pages 116–121.

Raimondi, F. (2006). *Model Checking Multi-Agent Systems.* PhD thesis, University College London.

Raimondi, F. and Lomuscio, A. (2004). Automatic verification of deontic interpreted systems by model checking via OBDD's. In de Mántaras, R. and Saitta, L., editors, *Proceedings of ECAI*, pages 53–57.

Rao, A. and Georgeff, M. (1991). Modeling rational agents within a BDI-architecture. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484.

Rosner, R. (1992). *Modular Synthesis of Reactive Systems.* PhD thesis, Weizmann Institute of Science.

Rovatsos, M., Rahwan, I., Fischer, F. A., and Weiß, G. (2005). Practical strategic reasoning and adaptation in rational argument-based negotiation. In *ArgMAS*, pages 122–137.

Sandholm, T. W. (1999). Distributed rational decision making. In Weiss, G., editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. The MIT Press, Cambridge, MA, USA.

Schewe, S. (2008). ATL* satisfiability is 2ExpTime-complete. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Part II (ICALP 2008), 6–13 July, Reykjavik, Iceland*, volume 5126 of *Lecture Notes in Computer Science*, pages 373–385. Springer-Verlag.

Schnoebelen, P. (2003). The complexity of temporal model checking. In *Advances in Modal Logics, Proceedings of AiML 2002.* World Scientific.

Schobbens, P. Y. (2004). Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93.

Shaw, P., Farwer, B., and Bordini, R. (2008). Theoretical and experimental results on the goal-plan tree problem (short paper). In *Proceedings of AAMAS'08*, pages 1379–1382.

Shoham, Y. and Tennenholz, M. (1992). On the synthesis of useful social laws for artificial agent societies. In *Proceedings of AAAI-92.*

Sistla, A. P. and Clarke, E. M. (1985). The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749.

Ståhl, I. (1972). *Bargaining Theory.* Stockholm School of Economics, Stockholm.

Stalnaker, R. (1994). On the evaluation of solution concepts. *Theory and Decision*, 37(1):49–73.

Stalnaker, R. (1996). Knowledge, belief and counterfactual reasoning in games. *Economics and Philosophy*, 12:133–163.

Su, K., Sattar, A., Governatori, G., and Chen, Q. (2005). A computationally grounded logic of knowledge, belief and certainty. In *Proceedings of AAMAS'05*, pages 149–156, New York, NY, USA. ACM Press.

Tang, Y. and Parsons, S. (2005). Argumentation-based dialogues for deliberation. In *Proc. of AAMAS*, pages 552–559.

Toda, S. (1989). On the computational power of pp and p. In *Proceedings of IEEE FOCS'89*, pages 514–519.

Valiant, L. G. (1979). The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201.

van Benthem, J. (2003). Rational dynamics and epistemic logic in games. In Vannucci, S., editor, *Logic, Game Theory and Social Choice III*, pages 19–23.

van der Hoek, W., Jamroga, W., and Wooldridge, M. (2005a). A logic for strategic reasoning. In *Proceedings of AAMAS'05*, pages 157–164.

van der Hoek, W., Roberts, M., and Wooldridge, M. (2005b). Social laws in alternating time: Effectiveness, feasibility and synthesis. *Synthese*.

van der Hoek, W., van Otterloo, S., and Wooldridge, M. (2004). Preferences in game logics. In *Proceedings of AAMAS-04*.

van der Hoek, W. and Wooldridge, M. (2002). Tractable multiagent planning for epistemic goals. In Castelfranchi, C. and Johnson, W., editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-02)*, pages 1167–1174. ACM Press, New York.

van der Hoek, W. and Wooldridge, M. (2003). Cooperation, knowledge and time: Alternating-time Temporal Epistemic Logic and its applications. *Studia Logica*, 75(1):125–157.

van Drimmelen, G. (2003). Satisfiability in Alternating-time Temporal Logic. In *Proceedings of LICS'2003*, pages 208–217. IEEE Computer Society Press.

van Otterloo, S. and Jonker, G. (2004). On Epistemic Temporal Strategic Logic. *Electronic Notes in Theoretical Computer Science*, XX:35–45. Proceedings of LCMAS'04.

van Otterloo, S. and Roy, O. (2005). Verification of voting protocols. Working paper, University of Amsterdam.

van Otterloo, S., van der Hoek, W., and Wooldridge, M. (2003). Knowledge as strategic ability. *Electronic Lecture Notes in Theoretical Computer Science*, 85(2).

van Otterloo, S., van der Hoek, W., and Wooldridge, M. (2004). Preferences in game logics. Preliminary version.

Vardi, M. Y. and Wolper, P. (1984). Automata theoretic techniques for modal logics of programs: (extended abstract). In *STOC '84: Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 446–456, New York, NY, USA. ACM.

Vardi, M. Y. and Wolper, P. (1986). An automata-theoretic approach to automatic program verification (preliminary report). In *Proceedings of the First Annual IEEE Symposium on Logic in Computer Science (LICS 1986)*, pages 332–344. IEEE Computer Society Press.

Vardi, M. Y. and Wolper, P. (1994). Reasoning about infinite computations. *Inf. Comput.*, 115(1):1–37.

Vollmer, H. (1999). *Introduction to Circuit Complexity.* Springer.

von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behaviour.* Princeton University Press: Princeton, NJ.

Walther, D., Lutz, C., Wolter, F., and Wooldridge, M. (2006). ATL satisfiability is indeed EXPTIME-complete. *Journal of Logic and Computation*, 16(6):765–787.

Weiss, G., editor (1999). *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence.* MIT Press: Cambridge, Mass.

Wilke, T. (1999). CTL+ is exponentially more succint than CTL. In *Proceedings of FST&TCS '99*, volume 1738 of *LNCS*, pages 110–121.

Wooldridge, M. (2000). *Reasoning about Rational Agents.* MIT Press : Cambridge, Mass.

Wooldridge, M. (2002). *An Introduction to Multi Agent Systems.* John Wiley & Sons.

Wooldridge, M. and Dunne, P. E. (2006). On the computational complexity of coalitional resource games. *Artif. Intell.*, 170(10):835–871.

Wozna, B. and Lomuscio, A. (2004). A logic for knowledge, correctness, and real time. In *Proceedings of CLIMA V*, Lecture Notes in Computer Science. Springer.