WESTERN SYDNEY UNIVERSITY
&
UNIVERSITÉ OF TOULOUSE

Logics for Strategic Reasoning and Collective Decision-Making

by
Guifei Jiang

A thesis submitted in fulfillment for the
degree of Doctor of Philosophy

in the
Artificial Intelligence Research Group (AIRG)
School of Computing, Engineering and Mathematics
&
Institut de Recherche en Informatique de Toulouse (IRIT)

September, 2016

Declaration of Authorship

I, Guifei Jiang, declare that this thesis titled, 'Logics for Strategic Reasoning and Collective Decision-Making' and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at the two universities.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at the two universities or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: [redacted]

Date: 01/09/2016

Thesis co-supervisors
Dongmo Zhang (WSU)
Laurent Perrusel (UT1)
Yan Zhang (WSU)

Author
Guifei Jiang

Logics for Strategic Reasoning and Collective Decision-Making

Abstract

Strategic decision-making is ubiquitous in everyday life. The analysis of game strategies has been a research theme in game theory for several decades since von Neumann and Morgenstern. Sophisticated models and analysis tools have been developed with wide applications in Economics, Management Science, Social Science and Computer Science, especially in the field of Artificial Intelligence. However, "*much of game theory is about the question whether strategic equilibria exist*", as Johan van Benthem, a world-leading logician and game-theorist, points out, "*but there are hardly any explicit languages for defining, comparing, or combining strategies*". Without such a facility it is challenging for computer scientists to build intelligent agents that are capable of strategic decision-making.

In the last twenty years, logical approaches have been proposed to tackle this problem. Pioneering work includes Game Logics, Coalition Logic and Alternating-time Temporal Logic (ATL). These logics either provide facilities for expressing and combining games or offer mechanisms for reasoning about strategic abilities of players. But none of them can solve the problem. The intrinsic difficulty in establishing such a logic is that reasoning about strategies requires combinations of temporal reasoning, counterfactual reasoning, reasoning about actions, preferences and knowledge, as well as reasoning about multi-agent interactions and coalitional abilities. More recently, a few new logical formalisms have been proposed by extending ATL with strategy variables in order to express strategies explicitly. However, most of these logics tend to have high computational complexity, because ATL introduces quantifications over strategies (functions), which leaves little hope of building any tractable inference system based on such a logic.

This thesis takes up the challenge by using a bottom-up approach in order to create a balance between expressive power and computational efficiency. Instead

of starting with a highly complicated logic, we propose a set of logical frameworks based on a simple and practical logical language, called *Game Description Language* (GDL), which has been used as an official language for General Game Playing (GGP) since 2005. To represent game strategies, we extend GDL with two binary prioritized connectives for combining actions in terms of their priorities specified by these connectives, and provide it with a semantics based on the standard state transition model.

To reason about the strategic abilities of players, we further extend the framework with coalition operators from ATL for specifying the strategic abilities of players. More importantly, a unified semantics is provided for both GDL- and ATL- formulas, which allows us to verify and reason about game strategies. Interestingly, the framework can be used to formalize the fundamental game-playing principles and formally derive two well-known results on two-player games: *Weak Determinacy* and *Zermelo's Theorem*. We also show that the model-checking problem of the logic is not worse than that of ATL*, an extension of ATL.

To deal with imperfect information games, we extend GDL with the standard epistemic operators and provide it with a semantics based on the epistemic state transition model. The language allows us to specify an imperfect information game and formalize its epistemic properties. Meanwhile, the framework allows us to reason about players' own as well as other players' knowledge during game playing. Most importantly, the logic has a moderate computational complexity, which makes it significantly different from similar existing frameworks.

To investigate the interplay between knowledge shared by a group of players and its coalitional abilities , we provide a variant of semantics for ATL with imperfect information. The relation between knowledge sharing and coalitional abilities is investigated through the interplay of epistemic and coalition modalities. Moreover, this semantics is able to preserve the desirable properties of coalitional abilities.

To deal with collective decision-making, we apply the approach of combining actions via their priorities for collective choice. We extend propositional logic with the prioritized connective for modelling reason-based individual and collective choices. Not only individual preferences but also aggregation rules can be expressed within this logic. A model-checking algorithm for this logic is thus developed to automatically generate individual and collective choices.

In many real-world situations, a group making collective judgments may assign individual members or subgroups different priorities to determine the collective judgment. We design an aggregation rule based on the priorities of individuals so as to investigate how the judgment from each individual affects group judgment in a hierarchical environment. We also show that this rule satisfies a set of plausible conditions and has a tractable computational complexity.

Acknowledgements

I wish to express my deepest gratitude to my principal supervisor A/Prof. Dongmo Zhang. He has always been extremely supportive and patient. Much was learned from him: how to perform high-quality research, how to make a decent presentation, and even how to speak better English. Above all, I thank him for the systematic guidance and great effort he put into training me in academic study.

I am also deeply grateful to my French supervisor Prof. Laurent Perrussel for his wonderful guidance and generous support throughout my doctoral study. He has always been highly insightful and considerate. I learnt great deal from him, not only in research, but in many other aspects. His willingness to help, his enthusiasm for research and his optimistic outlook on life, will inspire me through the rest of my days.

In addition to Dongmo and Laurent, I am fortunate and proud to have Prof. Yan Zhang on my supervisory panel. Many thanks for his advice and support in my research. Special thanks also to my supervisor Prof. Xiaojia Tang when I was in China, who introduced me to the field of game theory. She has been always supportive and concerned about my research.

I thank Prof. Sophie Pinchinat and A/Prof Sebastian Sardina for their willingness to assess the thesis as well as their valuable comments and recommendations to make it better.

I thank Prof. Emiliano Lorini from whom many things were learned, especially through working on our joint paper. His research ability and enthusiasm will remain a source of inspiration. I thank Prof. Wojtek Jamroga and A/Prof. Davide Grossi for their corrections and valuable suggestions regarding my work. Their rigorous attitude toward research will never be forgotten. I thank Prof. Ulle Endriss, A/Prof. Umberto Grandi, Prof. Andreas Herzig and Prof. Michael Thielscher for their insightful discussions about my research. I also thank Prof. Thomas Ågotnes, Prof. Johan van Benthem, Prof. Fenrong Liu, Dr. Jeremy Seligman, Dr. Haris Aziz and Dr. Nicholas Mattei for their insightful short courses or tutorials. Their views shaped my thoughts.

I thank all the teachers and colleagues from the three great research groups: AIRG, IRIT and CSLI, especially Dr. Dengji Zhao for his tremendous help and kind encouragement throughout my study, A/Prof. Yun Bai, Dr. Edward Caprin, Prof. Wu Chen, Dr. Jianying Cui, Dr. Li Dai, A/Prof. Sylvie Doutre, Dr. Benjun Gu, Prof. Meiyun Guo, Prof. Xiangdong He, Dr. Zhuhan, Jiang, Dr. Dave de Jonge, A/Prof. Minghui Ma, Dr. Filipo Perotto, A/Prof. Leanne Ryland, Dr. Jean-Marc Thévenin, Prof. Jing Wang, Prof. Weixing Zheng and Dr. Yi Zhou for their general guidance and support.

I thank Ms. Susan Henley, Ms. Virginie Mangion, Ms. Cuesta Michèle, Ms. Tracy Mills, Ms. Cheryl Murray, Ms. Marianne Oswald, Ms. Florence Thery, Ms. Linda Thornely and Mr. Nick Tothill for their kind help and support during my study. They were always ready to help with administrative or other non-academic problems. In particular, I thank Mr. Dennis Abbott for proofreading my thesis.

I thank my dearest friends: Zhu Pan, Hong Xia, Chathetana Illankoon, Hong Jiang, Xiaoyan Li, Shengli Ji, Junjun Deng, Chun Gao, Li Liu, Faustine Maffre, Jiefu Song and Zhanhao Xiao for their companion, caring and sharing, which made my life in Sydney and Toulouse memorable.

I thank Heng for his help and companion during my entire PhD study. He has been always there for me, cheering me up and standing by me through the good times and the bad. I am extremely fortunate to have him in my life.

I am deeply indebted to my family in ways far greater than I can express. Thanks to my parents for their selfless love and endless support, which always motivated me to work harder. I truly believe they are the greatest parents on Earth. Despite living in a remote village, they always cherished the belief that *knowledge changes destiny*. I cannot imagine how much patience, hard work and sacrifice they made to support their four children in obtaining a higher education and finding a bright future. Thanks to my three sisters for their enduring love and support, to help me concentrate on the study without allowing anything standing in my way.

Finally, I am grateful for the scholarships (IPRS&APA) from the Australian government, which supported my PhD research, and to the two universities for providing me with a wonderful learning environment. With the help of my supervisors, I have also profoundly benefited from the Cotutelle Program.

Guifei Jiang

Toulouse, July 2016

Contents

List of Figures

List of Tables

Abbreviations

AI	**A**rtificial **I**ntelligent
ATL	**A**lternating-time **T**emporal **L**ogic
EGDL	**E**pistemic **G**ame **D**escription **L**ogic
ET	**E**pistemic State **T**ransition
GDL	**G**ame **D**escription **L**anguage
GDR	Logic for **G**ame **D**escription and Strategic **R**easoning
LGD	**L**ogic for **G**ame **D**escription
MAS	**M**ulti-**A**gent **S**ystems
RCL	**R**eason-based **C**ollective Choice **L**ogic
ST	**S**tate **T**ransition

To my mum, Junhe Li, my dad, Shourong Jiang and my three sisters Guiling, Guiyun and Guiqing

Chapter 1

Introduction

This thesis introduces a set of logical formalisms to analyze strategic reasoning and collective decision-making so as to provide a logical foundation for game playing in Artificial Intelligence (AI) and Multi-Agent Systems (MAS). This chapter includes the motivation for this research, the literature review, the methods, the major contributions and the structure of the thesis.

1.1 Motivations

A *game* can be thought of as any strategic decision-making situation involving a group of *self-interested* agents, where the final outcome depends not only on the individual's own choice but also on the choices of others. For instance, when you play chess, the winner is determined not only by your own *strategy* (a general plan of actions) but also by the strategy of your opponent. *Game theory* has been used to predict the behaviour of *rational* agents and prescribe a plan of action that needs to be adopted. However, as van Benthem [2012] points out, "*much of game theory is about the question whether strategic equilibria exist. But there are hardly any explicit languages for defining, comparing, or combining strategies*" (p.96.).

This problem also challenges researchers of AI. The study of distributed and multi-agent systems typically deals with agents who have individual and possibly conflicting goals, and have a choice of actions to perform. This makes agents act strategically, attempting to choose their actions so as to guarantee their goals while facing other agents' actions. However, without such a facility for representing strategies explicitly, it is difficult for them to design intelligent agents that are capable of strategic decision making.

In the past two decades, formal logical analysis of strategic decision making has gathered momentum. Pioneering work includes Parikh and Pauly's Propositional Logic of Games, Kaneko and Nagashima's Predicate Logic of Games, Pauly's Coalition Logic (CL) and Alur et al.'s Alternating-time Temporal Logic (ATL). These logics either provide facilities for expressing and combing games or offer mechanisms for reasoning about strategic abilities of players. But none of them have solved van Benthem's problem. The intrinsic difficulty of establishing such a logic is that reasoning about game strategies requires combinations of temporal reasoning, counterfactual reasoning, reasoning about actions, preferences and knowledge, as well as reasoning about multi-agent interactions and coalitional abilities. To highlight this idea, let us consider an example.

Suppose that you and your friends are playing a game and this is the first time for all of you to play this game. The following questions might come to your mind:

- What's the game rule?
- What legal actions can I perform?
- What information is available to me and to others?
- Is there a strategy I can use to win?
  If yes, how to find such a strategy?
  If no, what strategy can I use to give myself a good chance to win?
- Can I collaborate with some of the other players to win?

These questions are even more important when we want to build an autonomous intelligent agent to play a game. Game-playing agents need to understand game rules and select their actions so as to guarantee their goals, even in face of other agents' actions. Basically, these questions involve game rules, game strategies, coalitional abilities, information and strategic reasoning. Accordingly, logical analysis of strategic decision-making needs to deal with the problems of

(1) How to describe the rules of a game?

(2) How to represent game strategies?

(3) How to specify strategic abilities of game players?

(4) How to clarify information and players' knowledge?

(5) How to model the strategic reasoning of game players?

These problems are closely interrelated. Without explicitly specifying game situations, without representing the game strategy under consideration, it is impossible to reason about the effects of these strategies. Therefore, we need a logical framework that is equipped with (i) *a language for describing the rules of a game, representing a game strategy and specifying strategic abilities of players*, (ii) *semantical structures for modelling the execution of a game*, and, more importantly, (iii) *an inference mechanism for the epistemic and strategic reasoning of game players.*

More recently, new logical formalisms have been proposed by extending ATL with strategy variables to express game strategies explicitly. However, most of these logics tend to have high computational complexity due to the introduction of quantifiers over strategies (functions) in ATL. For instance, to reason about strategies explicitly, Mogavero et al. [2014] propose Strategy Logic (SL), which strictly contains ATL, but the model checking problem for this logic becomes nonelementary. This leaves little hope for building any tractable inference systems based on such a logic. Therefore, we need a cautious and delicate way to establish such a logic so as to create a balance between expressive power and reasoning efficiency.

Moreover, playing games with imperfect information, such as Poker, poses a more intricate challenge to model the epistemic and strategic reasoning of players. In these games, players only have partial information of the current game state due to their limited observation powers or memory abilities, and they need to draw conclusions from their own knowledge about the current game state and about the knowledge of other players. If any logic is used in dealing with imperfect information games, such a logic must be able to reason about knowledge, time and actions at the same time. On the other hand, if we want to use this logic to build a game-playing agent that supports these reasoning mechanisms, the reasoning complexity of the logic must be in a range that the agent can manage, which poses another challenge to researchers of AI.

This thesis aims to address these challenges by establishing a set of logical frameworks to create a good balance between expressive power and computational efficiency.

1.2 Related Work

A number of logical formalisms have been developed to deal with different aspects of strategic decision-making. For the purposes of the thesis, this section discusses literature in terms of the following aspects: *game specification, game strategies, coalitional abilities, epistemic and strategic reasoning*. For a comprehensive overview of this topic, please refer to [Pacuit, 2015, van Benthem, 2014, van der Hoek and Pauly, 2006].

1.2.1 Game Specification

To specify a game situation, Parikh and Pauly's Propositional Logic of Games (GPL) treats a game as a program so that different games can be combined by program connectives [Parikh, 1985, Pauly and Parikh, 2003]. The benefit of this approach is that the effects of game playing can be specified by using propositional dynamic logic (PDL)-like inference mechanism. Yet GPL describes games as atomic objects, without specifying game structures. Kaneko and Nagashima [1996, 1997] propose another logic framework to capture the logical abilities of game players as well as the knowledge of a game situation, but to express the common knowledge concept explicitly, their base logic is an infinitary extension of classical predicate logic with very high reasoning complexity. De Giacomo et al. [2010] provide a different approach to specify a game structure based on situation calculus and ConGolog agent programming language. In their framework, a program specifies a game structure by making use of a background situation calculus action theory. Another approach is to treat games as interactive process models [van Benthem, 2002] and use dynamic logics to describe structures of extensive sequential games [van Benthem, 2001], as well as simultaneous games [van Benthem et al., 2008]. A more practical approach to specify a game is to use the so-called *Game Description Language* (GDL) [Genesereth et al., 2005]. This language is less expressive than the above mentioned game logics, but rich enough for describing any finite combinatorial games [Love et al., 2006]. Most importantly, GDL is designed as a machine-processable language with a tractable computational complexity. It has been used as an official language for General Game Playing (GGP) since 2005. However, as a purely game descriptive language, GDL does not provide the inference facility to reason about how a player derives unveiled information based on game rules.

1.2.2 Strategy Representation

To represent game strategies, a number of frameworks have been proposed. They can be categorized into three approaches. The first approach is to treat strategies as explicit first-order objects in which a strategy is a function from states or sequences of states to actions [Chatterjee et al., 2010, Mogavero, 2013, Mogavero et al., 2014, 2010]. Chatterjee et al. [2010] first introduce Strategy Logic (SL), a logic to use first-order quantifications over strategies in two-player turn-based games. Then Mogavero et al. extend SL to a more general framework for explicit reasoning about strategies in multi-player concurrent games. Yet this approach cannot model the internal structures of strategies, and thus it is difficult to show how to design a game strategy so as to achieve a goal state. Moreover, both SL and its extensions are highly undecidable [Mogavero et al., 2010]. The second approach is to express a game strategy as an action or a program so that simple strategies can be combined into more complicated strategies using PDL-like connectives [Ghosh, 2008, Ramanujam and Simon, 2006, 2008a,b, van Benthem, 2013, van Eijck, 2013]. Ramanujam and Simon [2006, 2008a,b] introduce a logic for reasoning about composite strategies in extensive form turn-based games. In their logic, players' strategies are treated as programs which are composed structurally by PDL-like connectives so as to ensure an outcome. Similarly, in [Ghosh, 2008, van Benthem, 2013, van Eijck, 2013] strategies are treated as partial transition relations and hence PDL provides a good framework to describe and reason about them. Yet this approach does not seem practical because the decision complexity for PDL and its variants is already rather high [Harel et al., 2000, Valiev, 1980]. Recently Zhang and Thielscher [2015a,b] introduce a third approach to represent strategies. They develop a modal logic, equipped with a variant of GDL to describe game strategies and a specific semantics by which formulas can be understood as move recommendations for a player. More importantly, they propose two preference operators, respectively called prioritized disjunction and prioritized conjunction, so as to combine simple strategies according to their priorities. However, their work can only model turn-based games and does not have the facility to reason about strategic abilities of players.

1.2.3 Reasoning about Coalitional Abilities

Many logics have been proposed to model coalitional abilities in perfect information games, mostly based on either Coalition Logic (CL) or Alternating-time Temporal Logic (ATL) [Alur et al., 2002, Pauly, 2002]. Both logics use coalition modalities to specify strategic abilities of players. In a nutshell, they use coalition operators of the form $[C]\phi$ to say coalition C (a set of players) has the ability to ensure temporal goal ϕ holds no matter what the other players do. However, these logics treat strategies implicitly through coalition modalities by using existential quantifiers to express players' strategic abilities, while description of strategies is not part of the logical language [van Benthem, 2012]. In other words, these logics do not model how a strategy is generated to ensure an outcome. To overcome this limitation, some work has been done to extend ATL with explicit expression of game strategies [Brihaye et al., 2009, Chatterjee et al., 2010, Mogavero et al., 2010, van der Hoek et al., 2005, Walther et al., 2007]. van der Hoek et al. [2005] introduce a logic for strategic reasoning based on ATL by treating strategies as first-class components of the language. The logic can not only reason about what coalitions can achieve, but also how they can achieve them. Based on this work, Walther et al. [2007] propose a variant of ATL with explicit names for strategies and develop a complete axiomatic system for this logic. However, as pointed out by Ramanujam and Simon [2008b] and van Eijck [2013], these extensions treat strategies as atomic objects without considering their internal structures. On the other hand, there is a price to pay for the rich expressiveness of these extensions. Because ATL introduces quantifiers over strategies, explicit representation of strategies incurs very high reasoning complexity. For instance, ATL with strategy contexts proposed by Brihaye et al. [2009] is undecidable in general [Troquard and Walther, 2012]. This leaves little hope of building any tractable inference system for representing and reasoning about game strategies based on such logics.

1.2.4 Reasoning about Information

To deal with imperfect information games, many logics, mostly epistemic extensions of ATL, SL and PDL, have been developed [Pacuit, 2014, Perea, 2014, van der Hoek and Pauly, 2006]. For instance, epistemic ATL-style logics can be used to specify and verify (epistemic) properties of multi-agent systems with imperfect information [Jamroga and van der Hoek, 2004, Schobbens, 2004, van der Hoek and Wooldridge, 2003]; Dynamic Epistemic Logic (DEL) takes imperfect information games as models for dynamic epistemic logic [Lorini and Schwarzentruber, 2010, van Benthem, 2001], and Epistemic Strategy Logic (ESL) extends temporal epistemic logic with operators that quantify over strategies so as to represent and reason about the knowledge that agents have of their own and other agents' strategies [Belardinelli, 2015, Huang and van der Meyden, 2014a,b]. Differently, Herzig and Troquard [2006] and Herzig and Lorini [2010] provide logical frameworks for reasoning about actions, agency and powers of agents and coalitions based on the logic of seeing-to-it-that (STIT) [Nuel et al., 2001]. Also Belle and Lakemeyer [2010] propose to reason about imperfect information games in the epistemic situation calculus. However, this approach does not seem practical due to the high complexity of the situation calculus. Recently, GDL has been extended to GDL-II so as to incorporate imperfect information games [Thielscher, 2010]. It can describe any extensive-form game with randomness and imperfect information [Thielscher, 2011]. Unfortunately, like GDL, GDL-II,as a purely descriptive language, is a tool only for describing the rules of an imperfect information game, but does not provide a facility for reasoning about how a player infers unveiled information based on game rules [Schiffel and Thielscher, 2011, 2014].

In brief, (i) logics to specify a game situation, such as GPL, GDL, are not designed for representing and reasoning about game strategies; (ii) logics to represent game strategies, such as SL and PDL-style logics, fail to specify a game situation explicitly; (iii) logics to describe games and represent strategies, such as the logic of Zhang and Thielscher [2015a,b], do not have the facility for reasoning about strategic abilities of players, and (iv) logics to reason about strategies, such as ATL and its extensions, fail to specify strategies or their structures explicitly. Thus, few logical frameworks can achieve all goals within a single logical formalism. On the other hand, logics to deal with imperfect information games either have the same expressiveness limitations inherited from their underlying logics, or have a rather high computational complexity, which are unlikely to be used for building practical agents in AI and MAS.

It should be noted that besides the above discussed work, more specific and technical literature will be mentioned in the relevant chapters.

1.3 Methods

This thesis uses logic-based methods to analyze strategic reasoning and collective decision-making. The following quote from Bacharach [1994] partially explains why logic is useful in games.

> Game theory is full of deep puzzles, and there is often disagreement about proposed solutions to them. The puzzlement and disagreement are neither empirical nor mathematical but, rather, concern the meanings of fundamental concepts ('solution', 'rational', 'complete information') and the soundness of certain argument $\cdots$. Logic appears to

> be an appropriate tool for game theory both because these conceptual obscurities involve notions such as reasoning, knowledge and counterfactuality which are part of the stock-in-trade of logic, and because it is a prime function of logic to establish the validity or invalidity of disputed arguments. (p.21.)

Specifically, there are several advantages to use logic-based methods for strategic reasoning and collective decision-making.

1. Formalisation is the first step towards automation. Logical formalism can be used together with tools and techniques developed in AI and computer science [Ågotnes et al., 2009, Wooldridge et al., 2007]. For instance,

    - As query languages for expressing properties of games or game strategies, checking whether a game or a game strategy has a property reduces to the model checking problems.
    - Reasoning about game strategies may be reduced to theorem proving.
    - Synthesising desirable properties of games corresponds to the satisfiability problem.

2. Logical analysis of strategic reasoning and collective decision making might also be of value in game theory and decision theory. Formal language can be used to make precise the important notions, such as knowledge and game strategies, that are often left informal or implicit in games, and rigorous mathematical models can be used to clarify confusions about the underlying assumptions of games. More importantly, they open the door for automated reasoning tools such as model checkers and theorem provers.

3. Logical formalisms are normally more succinct, compared to alternative methods. Representing game structures with concrete game models suffers from the *state explosion problem*. For instance, in Chess game, there

are almost 10^{30} states. Yet logical languages can provide a compact way for describing games. For instance, GDL is sufficient to describe any finite combinatorial game by encoding its rules [Love et al., 2006].

Nowadays there is a very active research area focused on using existing logical formalisms and developing new ones to analyze strategic reasoning and collective decision-making [Endriss, 2011, Pacuit, 2015, van Benthem, 2014].

1.4 Major Contributions

This thesis addresses the problem of modelling strategic reasoning and collective decision-making. The main contributions of this thesis can be summarized as follows:

1. Introduce a comprehensive logical formalism for game specifications, strategy representation and strategic reasoning of game players. Interestingly, this logic allows us to formalise van Benthem's game-oriented principles in multi-player games and formally derive two well-known results on two-player games: *Weak Determinacy* and *Zermelo's Theorem*. On the other hand, the model-checking problem of the logic is in PSPACE, which is not worse than that of ATL*.

2. Propose an epistemic extension of GDL to represent and reason about imperfect information games. The language allows us to represent the rules of an imperfect information game and formalize its epistemic properties. Meanwhile, the framework allows us to reason about a player's own as well as other players' knowledge during game playing. Most importantly, the model-checking problem of the framework is in Δ_2^p, which makes it significantly different from similar existing frameworks.

3. Provide a variant of semantics for ATL to investigate the interplay between knowledge shared by a group of agents and its coalitional abilities. Their relation is captured through the interplay of epistemic and coalition modalities. This result provides a partial answer to the question: which kind of group knowledge is required for a group to achieve their goals in the context of imperfect information [Herzig, 2015].

4. Introduce a modal logic for modelling reason-based individual and collective decision-making. Not only individual preferences but also collective choice rules can be built into this logic. This allows us to develop a model checking algorithm for this logic to automatically generate individual and collective choices, which is rarely achieved in the existing logics for social choice theory [Endriss, 2011].

5. Investigate how individual judgment affects group judgment in a hierarchical environment by designing a judgement aggregation rule based on the priority over individuals. This aggregation rule is specified by a set of plausible conditions and has a tractable computational complexity.

1.5 Outline of Chapters

Chapter 2 introduces the basic language GDL for describing game rules, and provides it with a semantics based on the state transition model. We then extend the basic logic with two binary prioritized connectives for representing game strategies. We also demonstrate with a generalised Gomuko game how to use the language to describe game rules, represent game strategies and formalize game properties.

Chapter 3 further extends the framework with coalition operators from ATL for specifying strategic abilities of players, and provides a unified semantics for both GDL- and ATL- formulas, which allows us to verify and reason about game strategies. We then use the framework to formalize the game-playing principles in multi-player games and derive two well-known results for two-player games: *Weak Determinacy* and *Zermelo's Theorem*. We also show the model-checking problem for the logic is in PSPACE.

Chapter 4 proposes an epistemic extension of GDL to deal with imperfect information games. The semantics is based on the epistemic state transition model. We demonstrate with a variant of a generalised Gomuko game how to use the language to represent the rules of an imperfect information game and formalize its epistemic properties, and, more importantly, how to use the logic to reason about a player's own as well as other players' knowledge during game playing. We also show the model-checking problem of the framework is Θ_2^p-hard, yet in Δ_2^p.

Chapter 5 investigates the interplay between knowledge shared by a group of agents and its coalitional abilities. To achieve this, we provide a variant of semantics for ATL based on the assumption of knowledge sharing within coalitions. We then investigate their relation through the interplay of epistemic and coalition operators. We also show this semantics preserves the desirable properties of coalitional abilities.

Chapter 6 explores collective decision-making by applying the prioritized connective introduced in Chapter 2 for collective choice. We first extend propositional logic with the prioritized connective so that a formula can express not only properties of alternatives but also individuals' priorities over the properties. We then define a set of collective choice rules within the same logic, which are specified by Arrowian conditions. We also develop a model-checking algorithm for the logic so as to automatically generate individual and collective choices.

Chapter 7 investigates how individual judgment affects group judgment in a hierarchical environment. We first provide a logic-based model for this situation by giving priorities over voters, and then define a lexicographic aggregation rule based on the priorities of voters. We also show this rule satisfies a set of plausible

Arrowian conditions and has a tractable computational complexity. We finally investigate the oligarchic property of this rule.

Chapter 8 summarizes the main results of this thesis and discusses some directions for future work.

Chapter 2

Game Description and Strategy Representation

This chapter addresses game specifications and strategy representation. Strategic reasoning will be discussed in Chapter 3. We first propose a logical framework for game specifications based on GDL. GDL is simple yet rich enough for describing any finite combinatorial games [Love et al., 2006]. It has been used as an official language for the annual General Game Playing (GGP) competitions since 2005 [Genesereth et al., 2005]. The semantics for the language is based on the standard state transition model. We then extend the basic framework with two binary prioritized connectives for strategy representation [Zhang and Thielscher, 2015b]. We also demonstrate with a running example that the framework allows us to specify game rules, formalize game properties as well as represent game strategies.

2.1 Game Description

We assume that all games we consider in this thesis are played in multi-agent environments. A *game signature* $\mathcal{S}$ is a triple $(N, \mathcal{A}, \Phi)$, where

- $N = \{r_1, r_2, \cdots, r_k\}$ is a nonempty finite set of agents.
- $\mathcal{A} = \bigcup_{r \in N} A^r$, where A^r consists of a nonempty finite set of *actions* performed by agent r and no action can be performed by two different agents, i.e., $A^{r_i} \cap A^{r_j} = \emptyset$ if $r_i \neq r_j \in N$.
- $\Phi = \{p, q, \cdots\}$ is a finite set of atomic propositions for specifying individual features of a game state.

Through the rest of the chapter, we will fix a game signature $\mathcal{S}$, and all concepts will be based on the same game signature unless otherwise specified.

2.1.1 State Transition Structures

We consider synchronous games where all players move simultaneously. These games take place in a multi-agent environment with one initial state, one or more terminal states and one or more winning states for each player. The execution model for these games is synchronous update: all players act together at every stage (although some actions could be "noops") and the environment updates only in response to the actions taken by the players. In its most abstract form, the underlying structures of these games may be specified by state transition frames defined as follows:

Definition 2.1. A *state transition (ST) frame* $\mathcal{F}$ is a tuple $(W, \overline{w}, T, L, U, g)$, where

- W is a nonempty set of *states*.
- $\overline{w} \in W$ is the *initial* state.
- $T \subseteq W$ is the set of *terminal* states.
- $L \subseteq W \times \mathcal{A}$ is a *legality* relation, describing the legal actions at each state.

- $U : W \times D \to W$ is an *update* function, where $D = \prod_{r \in N} A^r$ denote the set of joint actions, specifying state transitions for each joint action.

- $g : N \to 2^W$ is a *goal* function, specifying the winning states for each agent.

It should be noted that, to make the framework as general as possible, different from [Zhang and Thielscher, 2015b], we focus on synchronous games instead of turn-based games. The latter is a special case of the former by allowing a player only to do "noop", an action without effect, when it is not her turn. Given $d \in D$, let $d(r)$ be the individual action of agent r in the joint action d. For convenience, let $L(w) = \{a \in \mathcal{A} \mid (w, a) \in L\}$ be the set of all legal actions at state w.

Furthermore, to specify a particular synchronous game, we need to associate a state transition frame with a valuation function to address its domain-dependent information. In this way we obtain a state transition model.

Definition 2.2. A *state transition (ST) model* M is a pair $(\mathcal{F}, \pi)$ where

- $\mathcal{F}$ is an ST-frame;

- $\pi : W \to 2^\Phi$ is a standard valuation function.

We now define the set of all possible ways in which a game can develop as follows:

Definition 2.3. Given an ST-model $M = (\mathcal{F}, \pi)$, *a path* is a sequence of states and joint actions $\overline{w} \stackrel{d_1}{\to} w_1 \stackrel{d_2}{\to} \cdots \stackrel{d_e}{\to} w_e$ such that $e \geq 0$ and for any $j \in \{1, \cdots, e\}$,

1. $\{w_0, \cdots, w_{e-1}\} \cap T = \emptyset$ (that is, only the last state may be terminal.)

2. $d_j(r) \in L(w_{j-1})$ for any $r \in N$ (that is, any action that is taken must be legal.)

3. $w_j = U(w_{j-1}, d_j)$ (state update)

A path δ is *complete* if $w_e \in T$. Let $\mathcal{P}(M)$ denote the set of all complete paths in M. It follows that for any underlying ST-frame $\mathcal{F}$ of M, the set of all complete paths in $\mathcal{F}$, denoted by $\mathcal{P}(\mathcal{F})$, is just $\mathcal{P}(M)$. When M is fixed, we simply write $\mathcal{P}$. Given $\delta \in \mathcal{P}$, the states on δ are called *reachable states*. Let $\delta[j]$ denote the j-th reachable state of δ, $\theta(\delta, j)$ denote the joint action taken at stage j of δ, and $\theta_r(\delta, j)$ denote the action of agent r taken at stage j of δ. We write $\delta[0, j]$ for the *initial segment* of δ up to stage j. Finally, the length of a path λ, written $|\lambda|$, is defined as the number of joint actions.

To demonstrate the framework, we use as a running example a special family of mnk-games [van den Herik et al., 2002].

Example 2.1. *(mk**-Game**) An mk-game is a combinatorial game in which two players take turns in marking either a nought 'o' or a cross 'x' on an $m \times m$ board. The player who first gets k consecutive marks of her own symbol in a row (horizontally, vertically, or diagonally), will win this game.*

Obviously, an mk-game is a generalisation of Tic-Tac-Toe ($m = k = 3$) and Gomoku ($m = 19$ and $k = 5$).

To represent an mk-game in terms of the ST-model, we first describe the game signature, written $\mathcal{S}_{mk}$, as follows:

- $N_{mk} = \{\mathsf{x}, \mathsf{o}\}$.
- $A^r_{mk} = \{a^r_{i,j} \mid 1 \leq j, k \leq m\} \cup \{noop^r\}$, *where $a^r_{i,j}$ denotes the action that player r fills grid (i, j) with her symbol and $noop^r$ denotes that player r does action noop.*
- $\Phi_{mk} = \{p^r_{i,j}, turn(r) \mid r \in \{\mathsf{x}, \mathsf{o}\}$ *and* $1 \leq i, j \leq m\}$, *where $p^r_{i,j}$ represents the fact that grid (i, j) is filled with player r's symbol, and $turn(r)$ says that it is player r's turn now.*

We next specify the ST-frame for this game, written $\mathcal{F}_{mk}$, as follows:

- $W_{mk} = \{(t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{1,m}, c_{2,1}, \cdots, c_{m,m}) : t_{\mathsf{x}}, t_{\mathsf{o}} \in \{0,1\}$ & $c_{i,j} \in \{\square, \boxtimes, \boxdot\}$ *for* $1 \leq i, j \leq m\}$ *be the set of possible states, where* t_{x}, t_{o} *specify the turn taking and* $c_{i,j}$ *represents the fact that grid* (i,j) *is occupied by the cross* $\boxtimes$, *occupied by the nought* $\boxdot$ *or empty* $\square$.

- $\overline{w}_{mk} = (1, 0, \square, \cdots, \square)$.

- $g_{mk}(\mathsf{x}) = \{(t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}) : c_{i,j}, \cdots, c_{h,l} \in G_k \;\&\; c_{i,j} = \cdots = c_{h,l} = \boxtimes\}$, *and*

  $g_{mk}(\mathsf{o}) = \{(t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}) : c_{i,j}, \cdots, c_{h,l} \in G_k \;\&\; c_{i,j} = \cdots = c_{h,l} = \boxdot\}$,

  where

  $$\begin{aligned} G_k = &\{c_{i,j}, \cdots, c_{i,j+k-1} : 1 \leq i \leq m, 1 \leq j \leq m-k+1\} \\ &\cup \{c_{i,j}, \cdots, c_{i+k-1,j} : 1 \leq i \leq m-k+1, 1 \leq j \leq m\} \\ &\cup \{c_{i,j}, \cdots, c_{i+k-1,j+k-1} : 1 \leq i, j \leq m-k+1\} \\ &\cup \{c_{i,j}, \cdots, c_{i+k-1,j-k+1} : 1 \leq i \leq m-k+1, k \leq j \leq m\}^{[1]}. \end{aligned}$$

- $$\begin{aligned} T_{mk} = &\, g_{mk}(\mathsf{x}) \cup g_{mk}(\mathsf{o}) \\ &\cup \{(t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}) : c_{i,j} \in \{\boxtimes, \boxdot\} \text{ for } 1 \leq i, j \leq m\}. \end{aligned}$$

- *for all* $(t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}) \in W_{mk}$,

  - *for all* $a^r_{i,j} \in \mathcal{A}_{mk}$,
    $((t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}), a^r_{i,j}) \in L_{mk}$ *iff* $t_r = 1$ *and* $c_{i,j} = \square$;
  - *for* $noop^r \in \mathcal{A}_{mk}$, $((t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}), noop^r) \in L_{mk}$ *iff* $t_{-r} = 1$, *where* $-r$ *denotes the opponent of player* r.

- $U_{mk} : W_{mk} \times D_{mk} \to W_{mk}$ *is defined as follows: for all* $(t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}) \in W_{mk}$ *and for all* $\langle a^r_{i,j}, noop^{-r} \rangle \in D_{mk}$, *let*

  $$U_{mk}((t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}), \langle a^r_{i,j}, noop^{-r} \rangle) = (t'_{\mathsf{x}}, t'_{\mathsf{o}}, c'_{1,1}, \cdots, c'_{m,m})$$

[1] G_k is the set of all lines of k grids. For instance, on 3×3 board, G_3 has 8 elements.

such that $(t'_{\mathsf{x}}, t'_{\mathsf{o}}, c'_{1,1}, \cdots, c'_{m,m})$ *is the same as* $(t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m})$ *except its components* $t'_{\mathsf{x}}, t'_{\mathsf{o}}$ *and* $c'_{i,j}$ *which are updated as follows:* $t'_{\mathsf{x}} = t_{\mathsf{o}}$, $t'_{\mathsf{o}} = t_{\mathsf{x}}$ *and*

$$c'_{i,j} = \begin{cases} \boxtimes & \textit{if } c_{i,j} = \square \textit{ and } r = \mathsf{x} \\ \boxdot & \textit{if } c_{i,j} = \square \textit{ and } r = \mathsf{o} \\ c_{i,j} & \textit{otherwise} \end{cases}$$

Finally, for each state $w = (t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}) \in W_{mk}$, *let*

$$\pi_{mk}(w) = \{turn(r) : t_r = 1\} \cup \{p^{\mathsf{x}}_{i,j} : c_{i,j} = \boxtimes\} \cup \{p^{\mathsf{o}}_{i,j} : c_{i,j} = \boxdot\}.$$

Then let $M_{mk} = (\mathcal{F}_{mk}, \pi_{mk})$ *be the ST-model for this game.* ■

2.1.2 Logic for Game Description

In this subsection, we introduce a logical framework for specifying and reasoning about perfect information games. We call this framework *Logic for Game Description*, denoted as LGD.

2.1.2.1 The Language

Describing a game with the state transition game model is possible but not practical especially when modelling large games. For instance, in Chess game, there are almost 10^{30} states. We now introduce a variant[2] of GDL [Zhang and Thielscher, 2015b] for specifying games, which allows us to describe a game in a more compact way by encoding its rules.

Definition 2.4. The language, denoted by $\mathcal{L}_{LGD}$, consists of

- the finite set of atomic propositions $\Phi = \{p, q, \cdots\}$;

[2]GDL is a logical language in Zhang and Tielscher's work instead of an official program language for general game playing.

- pre-defined propositions: *initial*, *terminal*, $wins(\cdot)$, $legal(\cdot)$ and $does(\cdot)$;
- logical connectives: $\neg$ and $\wedge$;
- temporal operator: $\bigcirc$.

A formula φ in $\mathcal{L}_{LGD}$ is defined by the following BNF:

$$\varphi \ ::= \ p \mid initial \mid terminal \mid legal(a^r) \mid wins(r) \mid does(a^r) \mid \neg\varphi \mid \varphi \wedge \psi \mid \bigcirc\varphi$$

where $p \in \Phi$, $r \in N$ and $a^r \in A^r$.

Other connectives $\vee$, $\rightarrow$, $\leftrightarrow$, $\top$, $\bot$ are defined by $\neg$ and $\wedge$ in the standard way. Intuitively, *initial* and *terminal* specify the initial state and the terminal states of a game, respectively; $does(a^r)$ asserts that agent r takes action a at the current state; $legal(a^r)$ asserts that agent r is allowed to take action a at the current state; and $wins(r)$ asserts that agent r wins at the current state. The formula $\bigcirc\varphi$ means "φ holds in the next state".

To help the reader capture the intuition of the language, let us go back to the *mk*-games.

Example 2.1 (*continued*) *The rules of an mk-game can be naturally formulated by LGD-formulas in Figure 2.1 (where $r \in \{\mathsf{x}, \mathsf{o}\}$ and $-r$ represents r's opponent).*

Statement 1 says that all grids are empty in the initial state and player x has the first turn. Statements 2 and 3 specify the winning states for each player and the terminal states of the game, respectively. The player who succeeds in placing k respective marks in a horizontal, vertical or diagonal row wins the game, and the game ends if one player wins or all grids are filled. Statements 4 and 5 specify the preconditions of each action (legality). The player who has the turn can fill any empty grid. The other player can only do noop. Statement 6 is the combination of the frame axioms and the effect axioms [Reiter, 1991]: a grid is marked with

1. $initial \leftrightarrow turn(\mathsf{x}) \wedge \neg turn(\mathsf{o}) \wedge \bigwedge_{i,j=1}^{m} \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})$
2. $wins(r) \leftrightarrow \left(\begin{array}{c} \bigvee_{i=1}^{m}\bigvee_{j=1}^{m-k+1}\bigwedge_{l=0}^{k-1} p^r_{i,j+l} \vee \bigvee_{i=1}^{m-k+1}\bigvee_{j=1}^{m}\bigwedge_{l=0}^{k-1} p^r_{i+l,j} \\ \vee \bigvee_{i=1}^{m-k+1}\bigvee_{j=1}^{m-k+1}\bigwedge_{l=0}^{k-1} p^r_{i+l,j+l} \vee \bigvee_{i=1}^{m-k+1}\bigvee_{j=k}^{m}\bigwedge_{l=0}^{k-1} p^r_{i+l,j-l} \end{array}\right)$
3. $teminal \leftrightarrow wins(\mathsf{x}) \vee wins(\mathsf{o}) \vee \bigwedge_{i,j=1}^{m} (p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})$
4. $legal(a^r_{i,j}) \leftrightarrow \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}) \wedge turn(r)$
5. $legal(noop^r) \leftrightarrow turn(-r)$
6. $\bigcirc p^r_{i,j} \leftrightarrow p^r_{i,j} \vee (does(a^r_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})) \vee terminal$
7. $turn(r) \rightarrow \bigcirc\neg turn(r) \wedge \bigcirc turn(-r)$

FIGURE 2.1: An LGD description of an mk-game.

a player's symbol in the next state, if the player takes the corresponding action at the current state or the grid has been filled before or the game ends. The last formula specifies the turn-taking. Let Σ_{mk} be the set of rules 1-7. ■

2.1.2.2 The Semantics

The semantics for the language is based on the state transition model. With the temporal operator, a formula is interpreted with respect to two parameters: a complete path and a stage.

Definition 2.5. Let $M = (W, \overline{w}, T, L, U, g, \pi)$ be an ST-model. Given a complete path δ of M, a stage j on δ and a formula $\varphi \in \mathcal{L}_{LGD}$, we say φ is true (or satisfied) at j of δ under M, denoted by $M, \delta, j \models \varphi$, according to the following definition:

$M, \delta, j \models p$	iff	$p \in \pi(\delta[j])$
$M, \delta, j \models \neg\varphi$	iff	$M, \delta, j \not\models \varphi$
$M, \delta, j \models \varphi_1 \wedge \varphi_2$	iff	$M, \delta, j \models \varphi_1$ and $M, \delta, j \models \varphi_2$
$M, \delta, j \models initial$	iff	$\delta[j] = \overline{w}$

$$
\begin{array}{lll}
M, \delta, j \models \mathit{terminal} & \text{iff} & \delta[j] \in T \\
M, \delta, j \models \mathit{wins}(r) & \text{iff} & \delta[j] \in g(r) \\
M, \delta, j \models \mathit{legal}(a^r) & \text{iff} & a^r \in L(\delta[j]) \\
M, \delta, j \models \mathit{does}(a^r) & \text{iff} & \theta_r(\delta, j) = a^r \\
M, \delta, j \models \bigcirc\varphi & \text{iff} & \text{if } j < |\delta|, \text{ then } M, \delta, j+1 \models \varphi
\end{array}
$$

A formula φ is globally true through δ, denoted by $M, \delta \models \varphi$, if $M, \delta, j \models \varphi$ for any stage j of δ. A formula φ is *globally true* in an ST-model M, written $M \models \varphi$, if $M, \delta \models \varphi$ for all complete paths δ in M, that is, φ is true at every reachable state. A formula φ is *valid* in an ST-frame $\mathcal{F}$, denoted by $\mathcal{F} \models \varphi$, if it is globally true in every ST-model based on it. A formula φ is *valid* , denoted by $\models \varphi$, if it is valid in every ST-frame. Finally, let Σ be a set of formulas in $\mathcal{L}_{LGD}$, then M is a *model* of Σ if $M \models \varphi$ for all $\varphi \in \Sigma$.

Let us recall that M_{mk} is the ST-model for an $mk-$game and Σ_{mk} is the set of rules for the game. The following proposition shows that the framework provides a sound description for the mk-game.

Proposition 2.1. *M_{mk} is a model of Σ_{mk}.*

Proof. Given any complete path δ, any stage t of δ in M_{mk}, we need to verify that each rule is true at t of δ under M_{mk}.

Let us first verify Rule 1. Assume $M_{mk}, \delta, t \models \mathit{initial}$, then $\delta[t] = \overline{w}_{mk}$, i.e., $\delta[t] = (1, 0, \Box, \cdots, \Box)$. And by the definition of π_{mk}, $\mathit{turn}(\mathsf{x}) \in \pi_{mk}(\overline{w})$, $\mathit{turn}(\mathsf{o}) \notin \pi_{mk}(\overline{w})$, $p^r_{i,j} \notin \pi_{mk}(\overline{w})$ for any $r \in \{\mathsf{x}, \mathsf{o}\}$ and any $1 \leq i, j \leq m$. Thus, $M_{mk}, \delta, j \models \mathit{turn}(\mathsf{x}) \wedge \neg \mathit{turn}(\mathsf{o}) \wedge \bigwedge_{i,j=1}^{m} (\neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}))$.
Conversely, assume $M_{mk}, \delta, t \models \mathit{turn}(\mathsf{x}) \wedge \neg \mathit{turn}(\mathsf{o}) \wedge \bigwedge_{i,j=1}^{m} (\neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}))$, then by the definition of π_{mk}, $\delta[t] = (1, 0, \Box, \cdots, \Box)$, so $\delta[t] = \overline{w}_{mk}$. Thus, $M_{mk}, \delta, t \models \mathit{initial}$.

Rule 2 and Rule 3 are verified in a similar way of Rule 1.

Then we consider Rule 4, and Rule 5 is proved in a similar way. Assume $M_{mk}, \delta, t \models legal(a^r_{i,j})$ iff $a^r_{i,j} \in L_{mk}(\delta[t])$ iff $t(r) = 1$ and $c_{i,j}$ in $\delta[t]$ is empty $\Box$ (by the definition of L_{mk}) iff $turn(r) \in \pi_{mk}(\delta[t])$ and $p^r_{i,j} \notin \pi_{mk}(\delta[t])$ for any $r \in \{\mathsf{x}, \mathsf{o}\}$ (by the definition of π_{mk}) iff $M_{mk}, \delta, t \models \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}) \wedge turn(r)$.

We now verify Rule 6. Assume $M_{mk}, \delta, t \models terminal \vee p^r_{i,j} \vee (does(a^r_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}))$. We next prove by three cases.
(1) If $M_{mk}, \delta, t \models terminal$, then $t = |\delta|$, then it is trivial that $M_{mk}, \delta, t \models \bigcirc p^r_{i,j}$.
(2) If $M_{mk}, \delta, t \models p^r_{i,j}$, by the definition of U_{mk}, we have $M_{mk}, \delta, t+1 \models p^r_{i,j}$, so $M_{mk}, \delta, t \models \bigcirc p^r_{i,j}$.
(3) If $M_{mk}, \delta, t \models does(a^r_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})$, again by the definition of U_{mk}, we have $M_{mk}, \delta, t+1 \models p^r_{i,j}$, so $M_{mk}, \delta, t \models \bigcirc p^r_{i,j}$.
Thus, in all cases $M_{mk}, \delta, t \models \bigcirc p^r_{i,j}$.
Conversely, suppose $M_{mk}, \delta, t \not\models terminal \vee p^r_{i,j} \vee (does(a^r_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}))$, then $t < |\delta|$, $M_{mk}, \delta, t \not\models p^r_{i,j}$ and $M_{mk}, \delta, t \not\models does(a^r_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})$. Then by the definition of U_{mk}, $M_{mk}, \delta, t+1 \not\models p^r_{i,j}$, so $M_{mk}, \delta, t \not\models \bigcirc p^r_{i,j}$.

Finally we consider Rule 7. Assume $M_{mk}, \delta, t \models turn(r)$, then $t(r) = 1$. If $|\delta| = t$, then it is straightforward. If $|\delta| > t$, by the definition of U_{mk} and $\overline{w}_{mk}$, we have $t(-r) = 0$. And again by them we get that in $\delta[t+1]$, $t'(r) = 0$ and $t'(-r) = 1$, so $M_{mk}, \delta, t+1 \models \neg turn(r) \wedge turn(-r)$, so $M_{mk}, \delta, t \models \bigcirc(\neg turn(r) \wedge turn(-r))$. Thus, $M_{mk}, \delta, t \models \bigcirc \neg turn(r) \wedge \bigcirc turn(-r)$. $\Box$

2.1.3 Game Properties

In this section, we demonstrate the expressiveness and flexibility of the framework by showing how to use it to specify and verify game properties. To start with, the following validities describe interesting general (or domain-independent) properties of synchronous games.

Proposition 2.2. *For all φ, $\psi \in \mathcal{L}_{LGD}$,*

1. $\models \neg terminal \rightarrow \bigvee_{a^r \in A^r} does(a^r)$

2. $\models \neg(does(a^r) \wedge does(b^r))$ *for* $a^r \neq b^r \in A^r$

3. $\models does(a^r) \rightarrow legal(a^r)$

4. $\models \bigcirc(\varphi \rightarrow \psi) \rightarrow (\bigcirc\varphi \rightarrow \bigcirc\psi)$

5. $\models \bigcirc\top$

Proof. For any ST-model M, any complete path $\delta \in \mathcal{P}$ and any stage j of δ,

1. Assume $M, \delta, j \models \neg terminal$, then $|\delta| > j$, then $M, \delta, j \models does(\theta_r(\delta, j))$. And $\theta_r(\delta, j) \in A^r$, so $M, \delta, j \models \bigvee_{a^r \in A^r} does(a^r)$.

2. Suppose not for a contradiction that $M, \delta, j \models does(a^r) \wedge does(b^r)$, then $a^r = b^r = \theta_r(\delta, j)$ contradicting with the assumption $a^r \neq b^r \in A^r$.

3. Assume $M, \delta, j \models does(a^r)$, then $a^r = \theta_r(\delta, j)$. And by the definition of δ, $a^r \in L(\delta[j])$, so $M, \delta, j \models legal(a^r)$.

4. Assume $M, \delta, j \models \bigcirc(\varphi \rightarrow \psi)$ and $M, \delta, j \models \bigcirc\varphi$, we next show $M, \delta, j \models \bigcirc\psi$. Further assume $j < |\delta|$, then by assumption $M, \delta, j+1 \models \varphi \rightarrow \psi$ and $M, \delta, j+1 \models \varphi$, so $M, \delta, j+1 \models \psi$. Thus, $M, \delta, j \models \bigcirc\psi$.

It is straightforward for 5. □

Clauses 1 and 2 specify that there is only one action for each agent to be taken at each non-terminal state. Clause 3 states that an agent takes legal actions. The last two clauses indicate that the next operator is a KD-modality. The properties specified by these formulas hold for all ST-frames.

We next show that non-valid formulas in LGD can be used to classify game frames. Let us consider the following formulas.

1. $\not\models \neg terminal \rightarrow \neg \bigcirc initial$

2. $\not\models terminal \rightarrow \bigwedge_{a\in\mathcal{A}} \neg legal(a)$

3. $\not\models \bigvee_{r\in N} wins(r) \rightarrow terminal$

Formula 1 states that for any reachable non-terminal state the initial state is not its successor. Formula 2 says that at terminal states there is no legal action. The last one asserts that the game ends once one agent wins. It is easy to check that they are not valid in LGD. For instance, the single-agent state transition frame $\mathcal{F}_1$ depicted in Figure 2.2 is a simple countermodel for the first two formulas. Specifically, $\mathcal{F}_1$

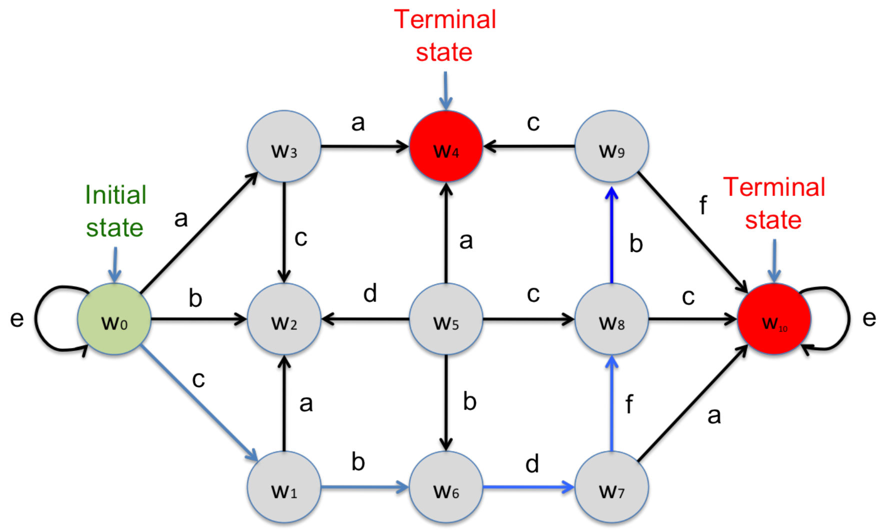

FIGURE 2.2: A state transition frame $\mathcal{F}_1$.

is a state transition frame with one initial state w_0 and two terminal (winning) states w_4 and w_{10}. In addition, the legality relation and the update function can be directly specified accordingly. Consider the following two complete paths in $\mathcal{F}_1$:

$$\delta_1 = w_0 \xrightarrow{e} w_0 \xrightarrow{a} w_3 \xrightarrow{a} w_4;\ \delta_2 = w_0 \xrightarrow{c} w_1 \xrightarrow{b} w_6 \xrightarrow{d} w_7 \xrightarrow{a} w_{10}$$

Given an arbitrary model M based on $\mathcal{F}_1$, we have that
$M, \delta_1, 0 \models \neg terminal \wedge \bigcirc initial$ and $M, \delta_2, 4 \models terminal \wedge legal(e)$.
Thus, $\mathcal{F}_1 \not\models \neg terminal \rightarrow \neg\bigcirc initial$ and $\mathcal{F}_1 \not\models terminal \rightarrow \bigwedge_{a\in\mathcal{A}} \neg legal(a)$.

Such formulas can be used to classify game frames as they correspond to different properties of frames. For instance, most board games satisfy the property specified by the first formula; two-player games normally have the property specified by the third formula. Specifically, let F_1 be the set of all ST-frames $\mathcal{F}$ satisfying the property that *$U(\delta[j], \theta(\delta, j)) \neq \overline{w}$ for any $\delta \in \mathcal{P}$ and any $j < |\delta|$*; F_2 be the set of all ST-frames $\mathcal{F}$ satisfying the property that *$L(w) = \emptyset$ for any reachable state $w \in T$*, and F_3 be the set of all ST-frames $\mathcal{F}$ satisfying the property that *for any reachable state $w \in W$, if $w \in g(r)$ for some $r \in N$, then $w \in T$*. Then we have the following characterization results.

Proposition 2.3. *For any ST-frame $\mathcal{F}$,*

1. $\mathcal{F} \in F_1$ *iff* $\mathcal{F} \models \neg terminal \to \neg \bigcirc initial$.
2. $\mathcal{F} \in F_2$ *iff* $\mathcal{F} \models terminal \to \bigwedge_{a \in \mathcal{A}} \neg legal(a)$.
3. $\mathcal{F} \in F_3$ *iff* $\mathcal{F} \models \bigvee_{r \in N} wins(r) \to terminal$.

Proof. 1. Assume $\mathcal{F} \in F_1$, then $U(\delta[j], \theta(\delta, j)) \neq \overline{w}$ for any $\delta \in \mathcal{P}$ and any $j < |\delta|$, then for any ST-model M based on $\mathcal{F}$, we have $M, \delta, j \models \neg \bigcirc initial$ for any $\delta \in \mathcal{P}$ and any $j < |\delta|$. And for $j = |\delta|$, we have $M, \delta, j \models terminal$. Thus, $M \models terminal \vee \neg \bigcirc initial$. So $\mathcal{F} \models terminal \vee \neg \bigcirc initial$. Conversely, assume $\mathcal{F} \models terminal \vee \neg \bigcirc initial$, then $M \models terminal \vee \neg \bigcirc initial$ for any ST-model M based on $\mathcal{F}$, then $M, \delta, j \models terminal \vee \neg \bigcirc initial$ for any $\delta \in \mathcal{P}$ and any j of δ. And for any $j < |\delta|$, we have $M, \delta, j \not\models terminal$, so for any $j < |\delta|$ $M, \delta, j \models \neg \bigcirc initial$, so $U(\delta[j], \theta(\delta, j)) \neq \overline{w}$ for any $\delta \in \mathcal{P}$ and any $j < |\delta|$. Thus, $\mathcal{F} \in F_1$.

2. Assume $\mathcal{F} \not\models terminal \to \bigwedge_{a \in \mathcal{A}} \neg legal(a)$, then there are some ST-model M based on $\mathcal{F}$, some complete path δ in M and some stage j on δ such that $M, \delta, j \models terminal$ but $M, \delta, j \not\models \bigwedge_{a \in \mathcal{A}} \neg legal(a)$, then $\delta[j] \in T$ and $a \in L(\delta[j])$ for some $a \in \mathcal{A}$, so there is reachable state $\delta[j] \in T$ such that $L(\delta[j]) \neq \emptyset$, so

$\mathcal{F} \notin F_2$. Conversely, assume $\mathcal{F} \models terminal \to \bigwedge_{a \in \mathcal{A}} \neg legal(a)$, then for any ST-model M based on $\mathcal{F}$, $M \models terminal \to \bigwedge_{a \in \mathcal{A}} \neg legal(a)$, then for any reachable state $w \in T$, $L(w) = \emptyset$, so $\mathcal{F} \in F_2$.

3 is proved in a similar way of 2. $\square$

In particular, we make the following observation about mk-games.

Observation 2.4. Let $\mathcal{F}_{mk}$ be the ST-frame for mk-games. Then

1. $\mathcal{F}_{mk} \models \neg terminal \to \neg \bigcirc initial$
2. $\mathcal{F}_{mk} \models \bigvee_{r \in N_{mk}} wins(r) \to terminal$
3. $\mathcal{F}_{mk} \not\models terminal \to \bigwedge_{a \in \mathcal{A}_{mk}} \neg legal(a)$

Proof. Statement 1 follows from the fact that $U_{mk}(w, d) \neq \overline{w}_{mk}$ for any reachable state $w \in W$ and $d \in D$. Statement 2 follows from the fact that $g_{mk}(r) \subseteq T$ for any $r \in N$. The last statement does not hold in M_{mk}. Consider a game situation where one player wins but the board is not completely filled. In this case, it is still legal for the player who has the turn to fill an empty grid. $\square$

It should be noted that an ST-frame specifies a class of synchronous games and by adding a valuation to address its domain-dependent information, an ST-model represents a particular game. Accordingly, valid formulas specify general (or domain-independent) properties of synchronous games of which we are interested, while globally true formulas prescribe specific properties of a particular game, such as the game rules in Figure 2.1 for an mk-game.

2.2 Strategy Representation

In this section, we consider the problem of how to represent a game strategy based on LGD. We first provide the notion of a game strategy, and then introduce an additional modality for representing a game strategy.

2.2.1 Strategy

A strategy for a player is a plan of actions telling this player how to play a game so as to achieve her goal. In game theory, there are two basic types of strategies: *memoryless strategies* and *memory-based strategies* [Osborne, 1994].

Before presenting their formal definitions, we first introduce some notations. Given an ST-model M and a complete path δ in M, any proper initial segment of δ is called *a history*. The set of all histories in M is denoted by $\mathcal{H}$. Let $\delta^w = \{\delta[j] \in W \mid 0 \leq j \leq |\delta|\}$ be the set of all the reachable states on δ. Then the set of all reachable states in M, denoted by $W^{\mathcal{P}}$, is the union of all sets δ^w, i.e., $W^{\mathcal{P}} = \bigcup_{\delta \in \mathcal{P}} \delta^w$.

We are now in the position to define the notion of a memoryless strategy for an agent.

Definition 2.6. Given an ST-model $M = (W, \overline{w}, T, L, U, g, \pi)$, a *memoryless strategy* for agent $r \in N$ in M is a total function f_r: $W^{\mathcal{P}} \backslash T \to A^r$ such that for every $w \in W^{\mathcal{P}} \backslash T$, $(w, f_r(w)) \in L$.

Intuitively, a memoryless strategy for an agent specifies a unique legal action for this agent at each non-terminal reachable state. On the other hand, a memory-based strategy for an agent requires that this agent should take the past states and actions into consideration when making a choice of actions.

Definition 2.7. Given an ST-model $M = (W, \overline{w}, T, L, U, g, \pi)$, a *memory-based strategy* for agent $r \in N$ in M is a total function f_r: $\mathcal{H} \to A^r$ such that for every $h \in \mathcal{H}$, $(last(h), f_r(h)) \in L$ where $last(h)$ denotes the last state of history h.

That is, a memory-based strategy for an agent assigns a unique action of the agent to all possible histories of the game. It should be noted that both memoryless strategies and memory-based strategies are agent-specific. That is, a strategy applies to one agent only. Hereafter, we focus on memoryless strategies and simply call them strategies; nevertheless, our approach can be generalized to the memory-based case.

2.2.2 Prioritized Connectives

Let us now introduce an additional modality to LGD so as to combine basic actions into a strategy. To give a rough idea of the composition, let us first consider a simple mk-game where $m = 5$ and $k = 3$.

After some practice or backward induction reasoning, we may find that the following simple idea can help player x to win:

(1) Fill the center.

(2) If filling any grid leads to win, fill it.

(3) Fill an empty grid next to her own symbol.

(4) Fill any grid.

(5) Try (1) first; if fails, try (2); if fails, try (3); if fails, do (4).

The simple actions (1)-(4) can be naturally formulated in LGD as follows:

1. $fill\_center^r =_{def} does(a^r_{3,3})$

2. $check^r =_{def} \bigvee_{i,j=1}^{5} (does(a^r_{i,j}) \wedge \bigcirc wins(r))$

3. $fill\_next^r =_{def} \bigvee_{i,j=1}^{5} \big(p^r_{i,j} \wedge (\, does(a^r_{i-1,j}) \vee does(a^r_{i,j-1}) \vee does(a^r_{i,j+1}) \vee does(a^r_{i+1,j}))\big)$[3]

4. $fill\_any^r =_{def} \bigvee_{i,j=1}^{5} does(a^r_{i,j})$

To win this game, a player needs to assign priorities over basic actions and then compose them into a winning strategy according to the priorities. Yet the underlying priorities over these actions in Clause (5) are beyond the expressiveness of LGD (also GDL). To represent these priorities, we introduce a binary prioritized disjunction, denoted by $\triangledown$, with the intention that *first take action a; if it is not available, then take action b, and so on.* Syntactically, we add the following formula formulation rule to the rules we already have in Definition 2.4:

$$\text{If } \varphi_1,\ \varphi_2 \in \mathcal{L}_{LGD}, \text{ then } \varphi_1 \triangledown \varphi_2 \text{ is a formula of } \mathcal{L}_{LGD}.$$

The formula $\varphi_1 \triangledown \varphi_2$ has to be read as "*try φ_1 first; if it is impossible, try φ_2*". In other words, when composing actions, an agent gives a higher priority to the action specified by φ_1 than to the action specified by φ_2.

We then define an abbreviation called the prioritized conjunction $\vartriangle$ as follows:

$$\varphi_1 \vartriangle \varphi_2 =_{def} (\varphi_1 \wedge \varphi_2) \triangledown \varphi_1$$

The formula means "*try both φ_1 and φ_2; if they are conflict, only try φ_1*". It is worth noting that the prioritized connectives are fully embedded in the logical language to make sure that context dependent strategies can be expressed, that is we may have the formulas like $\varphi \to (\varphi_1 \triangledown \varphi_2)$ and $\neg\varphi \to (\varphi_1 \vartriangle \varphi_2)$. Meanwhile we also allow the nesting of the prioritized connectives.

To interpret these formulas, we need an additional notation. Let $\delta[0, j]$ denote the *initial segment* of a complete path δ up to stage j. A complete path δ' that shares

[3] To avoid too much complexity, we ignore the cases when the indexes go over their range.

the same initial segment with δ up to stage j is denoted by $\delta[0,j] \sqsubseteq \delta'$. Given $\varphi \in \mathcal{L}_{LGD}$, let

$$\mathcal{P}(\varphi, \delta[0,j]) = \{\delta' \in \mathcal{P} \mid \delta[0,j] \sqsubseteq \delta' \text{ and } M, \delta', j \models \varphi\}$$

denote the set of all complete paths that share the same initial segment of δ up to stage j and satisfy φ at stage j. Now we add the following interpretation clause to Definition 2.5:

$$M, \delta, j \models \varphi_1 \triangledown \varphi_2 \quad \text{iff} \quad M, \delta, j \models \varphi_1, \text{ or } (\mathcal{P}(\varphi_1, \delta[0,j]) = \emptyset \text{ and } M, \delta, j \models \varphi_2)$$

To demonstrate the intuition behind the prioritized disjunction, let us consider the following example.

Example 2.2. *Assume that a complete path δ in a one-player state transition model M diverges at stage j due to executing different actions as illustrated in Figure 2.3. Consider a formula $does(\alpha)\triangledown does(b)$, which says "do action α first; if α is*

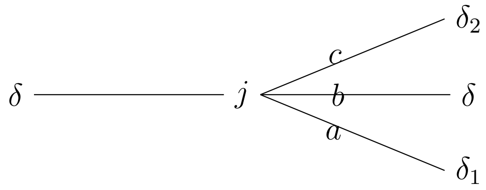

FIGURE 2.3: Example for $\triangledown$.

not doable, do b". We check whether it is true at stage j on δ. Since δ takes action b instead of α at j, we have $M, \delta, j \models \neg does(\alpha) \wedge does(b)$. And α is not doable at j, i.e., $\mathcal{P}(\alpha, \delta[0,j]) = \emptyset$. According to the semantics, $M, \delta, j \models does(\alpha)\triangledown does(b)$. Consider another formula $does(a)\triangledown does(b)$, which says "do action a first; if a is not doable, do b". Since action a can be done through δ_1 ($M, \delta_1, j \models does(a)$) and δ_1 shares the same initial segment of δ up to stage j (i.e., $\delta_1 \in \mathcal{P}(does(a), \delta[0,j])$). Thus a is doable at j. According to the semantics, $M, \delta, j \not\models does(a)\triangledown does(b)$. This is because δ picks up b even though a is doable at state j.

In addition, the following proposition provides the truth condition for the prioritized conjunction.

Proposition 2.5. $M, \delta, j \models \varphi_1 \vartriangle \varphi_2$ *iff*

$$M, \delta, j \models \varphi_1 \textit{ and } (\textit{if } \mathcal{P}(\varphi_1 \wedge \varphi_2, \delta[0, j]) \neq \emptyset, \textit{ then } M, \delta, j \models \varphi_2).$$

Proof.

Assume $M, \delta, j \models \varphi_1 \vartriangle \varphi_2$

iff $M, \delta, j \models (\varphi_1 \wedge \varphi_2) \triangledown \varphi_1$

iff $M, \delta, j \models (\varphi_1 \wedge \varphi_2)$, or $(\mathcal{P}(\varphi_1 \wedge \varphi_2, \delta[0, j]) = \emptyset$ and $M, \delta, j \models \varphi_1)$

iff $M, \delta, j \models \varphi_1$, and $(\mathcal{P}(\varphi_1 \wedge \varphi_2, \delta[0, j]) = \emptyset$ or $M, \delta, j \models \varphi_1 \wedge \varphi_2)$

iff $M, \delta, j \models \varphi_1$, and $(\mathcal{P}(\varphi_1 \wedge \varphi_2, \delta[0, j]) = \emptyset$ or $M, \delta, j \models \varphi_2)$

iff $M, \delta, j \models \varphi_1$, and (if $\mathcal{P}(\varphi_1 \wedge \varphi_2, \delta[0, j]) \neq \emptyset$, then $M, \delta, j \models \varphi_2$).

Thus, the result holds. □

To demonstrate the interpretation of the prioritized conjunction, let us go back to Example 2.2. Consider another formula $(does(a) \vee does(b)) \vartriangle does(b)$, which says that "try to make both $does(a) \vee does(b)$ and $does(b)$ true, i.e., do action b; if it is impossible, try to make $does(a) \vee does(b)$ true, i.e., do action a". It is easy to check that $M, \delta, j \models (does(a) \vee does(b)) \vartriangle does(b)$ as δ picks up b.

The ideas behind the prioritized connectives are similar to [Zhang and Thielscher, 2015b]. Basically, the prioritized disjunction "$\triangledown$" extends the choice of actions, such that if the first set of actions fails to apply, then a second one offers more options, and if that fails too then a third set may offer more options, and so on. Conversely, the prioritized conjunction "$\vartriangle$" narrows down the choice of actions such that if the first set of actions allows too many options, then a second set may be used to constrain these options, a third one may narrow down the options even further, and so on—up to the last set. These two connectives can be easily extended to take multiple arguments as follows:

$$\varphi_1 \triangledown \varphi_2 \triangledown \cdots \triangledown \varphi_m \ =_{def} ((\varphi_1 \triangledown \varphi_2) \ \triangledown \cdots) \ \triangledown \ \varphi_m$$

$$\varphi_1 \vartriangle \varphi_2 \vartriangle \cdots \vartriangle \varphi_m =_{def} ((\varphi_1 \vartriangle \varphi_2) \vartriangle \cdots) \vartriangle \varphi_m$$

From now on, we will use $\mathcal{L}^+_{LGD}$ to refer to the full language generated by the formation rules given by Definition 2.4 including the formation rule that allows the use of prioritized connectives.

2.2.3 Strategy Rule

We now discuss how to represent a game strategy in $\mathcal{L}^+_{LGD}$. Let us recall that a strategy specifies a unique action for an agent at each non-terminal reachable state. If a formula satisfies this condition, this formula can then be a syntactical representation of a strategy for that agent. This idea leads to the following definition.

Definition 2.8. Given an ST-model M and a formula $\varphi \in \mathcal{L}$, φ is a *strategy rule* for player r if for all $w \in W^{\mathcal{P}} \backslash T$, the set

$$A^r(\varphi, w) = \{a \in A^r : \exists \delta \in \mathcal{P} \exists j \in \mathbb{N}(M, \delta, j \models \varphi, \delta[j] = w \text{ and } \theta_r(\delta, j) = a)\}$$

is a singleton.

Similarly, a memory-based strategy rule for an agent is defined as follows:

Definition 2.9. Given an ST-model M and a formula $\varphi \in \mathcal{L}$, φ is *a memory-based strategy rule* for player r if for all $h \in \mathcal{H}$, the set

$$A^r(\varphi, h) = \{a \in A^r : \exists \delta \in \mathcal{P} \exists j \in \mathbb{N}(M, \delta, j \models \varphi, \delta[0, j] = h \text{ and } \theta_r(\delta, j) = a)\}$$

is a singleton.

It should be noted that our definition of strategy rules is different from that of Zhang and Thielscher [2015b]. They view a strategy as a set of possible moves, while we consider a strategy as an action plan that completely specifies the agent's future behavior when it is her turn, which coincides with the game-theoretical view. Obviously, not every formula $\varphi \in \mathcal{L}$ can be a strategy rule. In particular, we specify two types of formulas that have potentials to be strategy rules. We say φ is a *complete quasi-strategy rule* for player r if for all $w \in W^{\mathcal{P}} \backslash T$, $A^r(\varphi, w) \neq \emptyset$. Similarly, φ is a *deterministic quasi-strategy rule* for player r if for all $w \in W^{\mathcal{P}} \backslash T$ and $a, b \in A^r(\varphi, w)$ implies $a = b$ for all $a, b \in A^r$. Consequently, a complete and deterministic quasi-strategy rule is a strategy rule.

We next show how to create a strategy rule from non-strategy rules by combining a set of actions into a strategy using prioritized connectives. To this end, we start with some properties that can be used not only as a guideline to design a strategy, but also as a justification to check whether or not a formula is a strategy rule.

Proposition 2.6. *Given an ST-model M, for each player $r \in N$ and for all $\varphi_1, \varphi_2 \in \mathcal{L}^+_{LGD}$,*

1. *if either φ_1 or φ_2 is a complete quasi-strategy rule for r, so is $\varphi_1 \triangledown \varphi_2$.*
2. *φ_1 is a complete quasi-strategy rule for r iff $\varphi_1 \vartriangle \varphi_2$ is a complete quasi-strategy rule for r.*
3. *If both φ_1 and φ_2 are deterministic quasi-strategy rules for r, so is $\varphi_1 \triangledown \varphi_2$.*
4. *If φ_1 is a deterministic quasi-strategy rule for r, so is $\varphi_1 \vartriangle \varphi_2$.*

Proof. 1. Suppose not for a contradiction that $\varphi_1 \triangledown \varphi_2$ is not complete. Then there is some $w \in W^{\mathcal{P}} \backslash T$ such that $A^r(\varphi_1 \triangledown \varphi_2, w) = \emptyset$, so for all δ for all stage j if $\delta[j] = w$ then $M, \delta, j \not\models \varphi_1 \triangledown \varphi_2$, so $M, \delta, j \not\models \varphi_1$ and $(\mathcal{P}(\varphi_1, \delta[0, j]) \neq \emptyset$ or $M, \delta, j \not\models \varphi_2)$. Since $M, \delta, j \not\models \varphi_1$ contradicts with $\mathcal{P}(\varphi_1, \delta[0, j]) \neq \emptyset$, so $M, \delta, j \not\models$

φ_1 and $M, \delta, j \not\models \varphi_2$. Thus, $A^r(\varphi_1, w) = \emptyset$ and $A^r(\varphi_2, w) = \emptyset$, contradicting with the assumption, so $\varphi_1 \triangledown \varphi_2$ is a complete quasi-strategy rule for r.

3. Suppose not for a contradiction that $\varphi_1 \triangledown \varphi_2$ is not deterministic, then there is some $w \in W^{\mathcal{P}} \backslash T$ such that $a \in A^r(\varphi_1 \triangledown \varphi_2, w)$ and $b \in A^r(\varphi_1 \triangledown \varphi_2, w)$ for $a \neq b \in A^r$, then
(there are δ and j such that $\delta[j] = w$, $M, \delta, j \models \varphi_1 \triangledown \varphi_2$ and $\theta_r(\delta, j) = a$) and
(there are δ' and j' such that $\delta'[j'] = w$, $M, \delta', j' \models \varphi_1 \triangledown \varphi_2$ and $\theta_r(\delta', j') = a$), so
$M, \delta, j \models \varphi_1$ or $(\mathcal{P}(\varphi_1, \delta[0, j]) = \emptyset$ and $M, \delta, j \models \varphi_2)$ and
$M, \delta', j' \models \varphi_1$ or $(\mathcal{P}(\varphi_1, \delta'[0, j']) = \emptyset$ and $M, \delta', j' \models \varphi_2)$.
Then there are four cases, and it is not hard to get a contradiction for each case.

Clauses 2 and 4 are straightforward. □

This result can be generalized to multiple arguments as follows:

Proposition 2.7. *Given an ST-model M, for each player $r \in N$ and for all $\varphi_1, \cdots, \varphi_n \in \mathcal{L}^+_{LGD}$,*

1. *if for some i $(1 \leq i \leq n)$ φ_i is a complete quasi-strategy rule for r, so is $\varphi_1 \triangledown \cdots \triangledown \varphi_n$.*
2. *φ_1 is a complete quasi-strategy rule for r iff $\varphi_1 \vartriangle \cdots \vartriangle \varphi_n$ is a complete quasi-strategy rule for r.*
3. *If for all i $(1 \leq i \leq n)$ φ_i is a deterministic quasi-strategy rule for r, so is $\varphi_1 \triangledown \cdots \triangledown \varphi_n$.*
4. *If φ_1 is a deterministic quasi-strategy rule for r, so is $\varphi_1 \vartriangle \cdots \vartriangle \varphi_n$.*

Proof. 1. We prove this by induction on n.

- For $n = 1$. It is trivial by assumption since $i = 1$;

- For $n = l + 1$. Given any $w \in W^{\mathcal{P}} \backslash T$, by the induction hypothesis that $\varphi_1 \triangledown \cdots \triangledown \varphi_l$ is a complete quasi-strategy rule for r, we have there is some $a \in A^r(\varphi_1 \triangledown \cdots \triangledown \varphi_l, w)$, then there are some complete path δ and some stage j on δ such that $M, \delta, j \models \varphi_1 \triangledown \cdots \triangledown \varphi_l$, $\delta[j] = w$ and $\theta_r(\delta, j) = a$, then $M, \delta, j \models \varphi_1 \triangledown \cdots \triangledown \varphi_{l+1}$, $\delta[j] = w$ and $\theta_r(\delta, j) = a$. So $a \in A^r(\varphi_1 \triangledown \cdots \triangledown \varphi_{l+1}, w)$.

Thus, $\varphi_1 \triangledown \cdots \triangledown \varphi_n$ is also a complete quasi-strategy rule for r.

2. The direction from the left to the right follows from Proposition 2.6.2, and the other direction is straightforward.

3. Assume for all i $(1 \leq i \leq n)$ φ_i is a deterministic quasi-strategy rule for r, then by Proposition 2.6.3, $\varphi_1 \triangledown \varphi_2$ is deterministic. And by assumption φ_3 is a deterministic quasi-strategy rule for r, so is $\varphi_1 \triangledown \varphi_2 \triangledown \varphi_3$ by Proposition 2.6.3. Repeat this process, we obtain that $\varphi_1 \triangledown \cdots \triangledown \varphi_n$ is a deterministic quasi-strategy rule for r.

4. It follows from Proposition 2.6.4. □

Statement 1 provides us an easy way of generating a complete quasi-strategy rule: *create a trivial complete quasi-strategy rule first and then combine it with other quasi-strategy rules using the prioritized disjunction.* Statement 2 tells us that once we get a complete quasi-strategy rule, we can further refine the quasi-strategy rule targeting more specific property, say deterministic, using the prioritized conjunction without losing its completeness. Statement 3 shows us another feasible way of generating a strategy rule: instead of creating a complete quasi-strategy rule then refining it into a deterministic one, we can *devise a set of specific deterministic quasi-strategy rules first and then combine them into a complete one* [Zhang and Thielscher, 2015b].

To demonstrate how they work, let us go back to the 53-game at the beginning of this section.

(1) Fill the center.

(2) If filling any grid leads to win, fill it.

(3) Fill an empty grid next to her own symbol.

(4) Fill any grid.

(5) Try (1) first; if fails, try (2); if fails, try (3); if fails, do (4).

The actions (1)-(4) are represented in $\mathcal{L}_{LGD}$ (also in $\mathcal{L}^{+}_{LGD}$) as follows: for $r \in \{\mathsf{x}, \mathsf{o}\}$,

1. $fill\_center^r =_{def} does(a^r_{3,3})$
2. $check^r =_{def} \bigvee_{i,j=1}^{5} (does(a^r_{i,j}) \wedge \bigcirc wins(r))$
3. $fill\_next^r =_{def} \bigvee_{i,j=1}^{5} \big(p^r_{i,j} \wedge (\, does(a^r_{i-1,j}) \vee does(a^r_{i,j-1}) \vee does(a^r_{i,j+1}) \vee does(a^r_{i+1,j}))\big)$[4]
4. $fill\_any^r =_{def} \bigvee_{i,j=1}^{5} does(a^r_{i,j})$

With the prioritized connectives, Clause (5) can be naturally described in $\mathcal{L}^{+}_{LGD}$ as follows:

5. $combined^r =_{def} fill\_center^r \triangledown check^r \triangledown fill\_next^r \triangledown fill\_any^r$

It is easy to see that formula $combined^r$ can be satisfied in any state when it is r's turn (due to $fill\_any^r$). However, it is not a strategy rule because it may suggest more than one actions in one state. To make it deterministic, we need the following technical treatment.

Let $B = \{(i,j) : 1 \leq i,j \leq 5\}$ be the game board and $\prec$ be the lexicographic order on B, i.e., $(1,1) \prec (1,2) \prec \cdots \prec (1,5) \prec (2,1) \prec \cdots \prec (5,5)$. For each grid

[4] To avoid too much complexity, we ignore the cases when the indexes go over their range.

$\xi \in B$, let $c^r_\xi = \bigvee_{(i,j)\prec\xi} does(a^r_{i,j})$, which represents the idea of player r to fill any grid from $(1,1)$ up to ξ. We let

$$\widehat{S}^r_{53} =_{def} \big((fill\_center^r \triangledown check^r \triangledown fill\_next^r \triangledown fill\_any^r) \vartriangle c^r_{(5,5)} \vartriangle \cdots \vartriangle c^r_{(1,1)}\big)$$

It is not hard to check that $\widehat{S}^r_{53}$ specifies a unique action for agent r at each state where it is her turn. To obtain a strategy rule, we also need to consider the states where it is not her turn by simply allowing agent r to take action *noop*. Let

$$S^r_{53} =_{def} (turn(r) \to \widehat{S}^r_{53}) \wedge (\neg turn(r) \to does(noop^r)) \tag{2.1}$$

Then we make the following observation.

Observation 2.8. For any player $r \in \{\mathsf{x},\mathsf{o}\}$, S^r_{53} is a strategy rule for player r.

Proof. For any non-terminal reachable state w, if it is player r's turn, then $\widehat{S}^r_{53}$ specifies a unique action for r at w; otherwise, i.e., it is not her turn, then she does action *noop* at w. □

To illustrate that the framework can be used to design more complicated strategies, let us consider another game: Tic-tac-toe (m = k = 3). We need some additional actions. For $r \in \{\mathsf{x},\mathsf{o}\}$,

- Fill a corner.

  $fill\_corner^r =_{def} \bigvee_{i\in\{1,3\}} \bigvee_{j\in\{1,3\}} does(a^r_{i,j})$

- Fill an edge.

  $fill\_edge^r =_{def} \bigvee_{i\in\{1,3\}} (does(a^r_{i,2}) \vee does(a^r_{2,i}))$

- If the opponent makes to win by filling a grid in the next state, then the player blocks her by filling the grid now.

  $block^r =_{def} \bigvee_{i=1}^3 \bigvee_{j=1}^3 (\bigcirc(does(a^{-r}_{i,j}) \wedge \bigcirc wins(-r)) \wedge does(a^r_{i,j}))$

- A player fills the opposite corner whenever a corner is filled by her opponent.

  $fill\_Ocorner^r =_{def} \bigvee_{i\neq j\in\{1,3\}}((p^{-r}_{i,i} \wedge does(a^r_{j,j})) \vee (p^{-r}_{j,j} \wedge does(a^r_{j,i})))$

- A player fills a besieged corner to avoid the opponent to form a double threat.

  $fill\_Bcorner^r =_{def} \bigvee_{i\neq j\in\{1,3\}}(p^{-r}_{i,2} \wedge p^{-r}_{2,j} \wedge does(a^r_{i,j}))\vee \bigvee_{i\in\{1,3\}}(p^{-r}_{i,2} \wedge p^{-r}_{2,i} \wedge does(a^r_{i,i}))$

- A player fills an edge when the opponent fills a pair of opposite corners.

  $fill\_Cedge^r =_{def} ((p^{-r}_{1,1} \wedge p^{-r}_{3,3}) \vee (p^{-r}_{1,3} \wedge p^{-r}_{3,1})) \wedge fill\_edge^r$

Based on these, we are now in the position to design a strategy rule for player x filling the center as the opening action.

$$S^{\mathsf{x}}_{33} =_{def} (turn(\mathsf{x}) \to \widehat{S}^{\mathsf{x}}_{33}) \wedge (\neg turn(\mathsf{x}) \to noop^{\mathsf{x}}) \tag{2.2}$$

where

$\widehat{S}^{\mathsf{x}}_{33} =_{def} (fill\_center^{\mathsf{x}} \triangledown check^{\mathsf{x}} \triangledown block^{\mathsf{x}} \triangledown fill\_corner^{\mathsf{x}} \triangledown fill\_any^{\mathsf{x}}) \vartriangle c^{\mathsf{x}}_{(3,3)} \vartriangle \cdots \vartriangle c^{\mathsf{x}}_{(1,1)}$

Intuitively, the strategy rule S^{x}_{33} specifies player x to take actions when it is her turn in such a prioritized way: first fill the center; if this fails, then try to find chances to win; if impossible, try to block her opponent; if no threat exists, then fill a corner; if all fail, then fill any available grid. Note that action $fill\_any^{\mathsf{x}}$ guarantees that there should be at least one action for player x to take at all non-terminal reachable states when it is her turn, and, on the other hand, the $\vartriangle$ formulas refine these actions to a unique one, i.e., the smallest one. By taking action *noop* for player x at all non-terminal reachable states when it is not her turn, we make the following observation.

Observation 2.9. S^{x}_{33} is a strategy rule for player x.

Meanwhile, a strategy rule for player o is given as follows:

$$S_{33}^{\mathsf{o}} =_{def} (turn(\mathsf{o}) \rightarrow \widehat{S}_{33}^{\mathsf{o}}) \wedge (\neg turn(\mathsf{o}) \rightarrow does(noop^{\mathsf{o}})) \tag{2.3}$$

where

$\widehat{S}_{33}^{\mathsf{o}} =_{def} (check^{\mathsf{o}} \triangledown block^{\mathsf{o}} \triangledown\ fill\_center^{\mathsf{o}} \triangledown\ fill\_Cedge^{\mathsf{o}} \triangledown\ fill\_Bcorner^{\mathsf{o}} \triangledown\ fill\_Ocorner^{\mathsf{o}}$ $\triangledown\ fill\_corner^{\mathsf{o}} \triangledown\ fill\_any^{\mathsf{o}}) \vartriangle c_{(3,3)}^{\mathsf{o}} \vartriangle \cdots \vartriangle c_{(1,1)}^{\mathsf{o}}$

That is, the strategy rule S_{33}^{o} specifies player o to take actions in such a way: when it is her turn, first check to win; if it fails, block to avoid an immediate loss; if it fails again, fill the center; if it is impossible, then fill an edge when the opponent fills the opposite corners; if this fails, fill a besieged corner to avoid a double threat; if there is no besieged corner, fill an opposite corner; if this is still unavailable, then just fill a corner; if all fail, fill any available grid; when it is not her turn, player o simply does action *noop*.

Observation 2.10. S_{33}^{o} is a strategy rule for player o.

It should be noted that the conjuncts for no-ops in strategy rules 2.1-2.3 are only added for asynchronous games, which shows the proposed framework is general enough to treat asynchronous games as special cases by allowing a player only to do "noop" when it is not her turn. Moreover, we have demonstrated that the framework is able to represent game strategies for two specific *mk*-games, namely 53-game and 33-game; nevertheless, it is expressive to describe more complicated strategies for more complicated games, such as Gomoku game (m = 15, k = 5).

2.3 Summary

In this chapter, we have presented a logical framework by extending GDL with prioritized connectives for strategy representation. With a running example, we have demonstrated that the language allows us to specify game rules and formalize game properties, as well as represent game strategies. In the next chapter, we will reason about strategic abilities of game players and provide a constructive approach to show whether a game can be forced in a draw or certain players can cooperate to win. In particular, we will show that the strategy rule designed for 53-game can guarantee the first player to win, and the strategy rule for each player in 33-game actually specifies a no-losing strategy for the corresponding player.

Besides the related work discussed in Section 1.2, the following is also worth mentioning. There is some recent work to use a Turing machine based model for strategy representation and composition [Gelderie, 2012, 2013, 2014]. It mainly studies the representation of winning strategies in infinite games as Turing machines. The idea of composing actions via priorities is first proposed by Zhang and Thielscher [2015b], but there are three essential differences between their work and ours. Firstly, their definition of the prioritized connectives is based on the semantics of strategies rather than on the semantics of the logic. Strictly speaking, their prioritized connectives are not part of their logical language. However, ours are part of the logical language. Secondly, we define these connectives as binary operators, while theirs are multiple tuple operators. Thirdly, in the next chapter, we will further extend this framework for reasoning about strategic abilities of players, which is not involved in their work.

Chapter 3

Coalitional Abilities and Strategic Reasoning

Chapter 2 introduces a logical framework for game specifications and strategy representation. It does not reason about game strategies and specify how different agents can cooperate to achieve a desirable goal. This chapter addresses this issue by modelling coalitional abilities and strategic reasoning. We first present a unified framework by further extending the language for strategy representation with coalition operators from ATL [Alur et al., 2002]. We then show that this framework allows us to formalise van Benthem's game-oriented principles in multi-player games, and formally derive Weak Determinacy and Zermelo's Theorem for two-player games. We also demonstrate how to use this framework to verify a winning/no-losing strategy and reason about coalitional abilities. Finally we show that the model-checking problem of the logic is in PSPACE, which is not worse than the model-checking problem for ATL*, an extension of ATL.

3.1 Coalitional Abilities

In games, an agent might cooperate with others in order to bring about a desirable state of affairs. In this section, we begin with the notion of an *effectivity function* which aims at capturing explicitly the power that can be obtained by a group of agents if they form a coalition, and then introduce additional operators so as to specify and reason about coalitional abilities.

3.1.1 Effectivity Functions

Effectivity functions model the distribution of power among individuals and groups of individuals. They have been studied extensively in game theory and social choice theory [Abdou and Keiding, 2012, Peleg, 1997]. The following definition is borrowed from [van der Hoek and Pauly, 2006].

Definition 3.1. Given a nonempty finite set of players N and a nonempty set of states S, an *effectivity function* is any function E: $2^N \to 2^{2^S}$ such that (i) for any $C \subseteq N$, $\emptyset \notin E(C)$, and (ii) for any $C \subseteq N$, $S \in E(C)$.

The function E associates to every group of players the sets of outcomes for which the group is effective; coalition C is *effective* for $X \subseteq S$ if it can bring about a state in X. Intuitively, condition (i) states that no coalition C can enforce the falsity, and condition (ii) says that every coalition can ensure the truth.

In most situations, effectivity functions are expected to satisfy some properties [Pauly, 2001, 2002].

Outcome Monotonicity. *For all $C \subseteq N$ and $X \subseteq X' \subseteq S$, if $X \in E(C)$, then $X' \in E(C)$.* This requires that if a coalition is effective for a subset of states, then it is also effective for its supersets, since a superset of states places fewer constraints on a coalition's ability.

Coalition Monotonicity. *For all* $C \subseteq C' \subseteq N$*,* $E(C) \subseteq E(C')$*.* The property indicates that if a coalition can achieve some goal, then its superset can achieve this goal as well.

C**-Regularity.** *For all* $C \subseteq N$*, if* $X \in E(C)$*, then* $S \backslash X \notin E(N \backslash C)$*.* This means that it is impossible for a coalition and its complementary set to enforce inconsistency.

C**-Maximality.** *For all* $C \subseteq N$*, if* $S \backslash X \notin E(N \backslash C)$*, then* $X \in E(C)$*.* This means that for every coalition C, either C itself or its complementary set can at least bring about something. Take a two-player game over $S = \{win_1, win_2\}$ for an example: $\{1\}$-maximality states that the game is *determined*: if one player does not have a winning strategy, then the other player does.

Superadditivity. *For all* X_1*,* $X_2 \subseteq S$ *and* C_1*,* $C_2 \subseteq N$ *with* $C_1 \cap C_2 = \emptyset$*, if* $X_1 \in E(C_1)$ *and* $X_2 \in E(C_2)$*, then* $X_1 \cap X_2 \in E(C_1 \cup C_2)$*.* The property specifies that disjoint coalitions can combine their strategies to achieve more.

3.1.2 Coalition Operators

Let us now introduce two additional modalities to represent and reason about coalitional abilities captured by effectivity functions. We extend the language $\mathcal{L}^{+}_{LGD}$ with two coalition operators, denoted by $[C]$ and $[\![C]\!]$, which intend to specify coalition enforcement. Syntactically, we add the following formulation rule to the language $\mathcal{L}^{+}_{LGD}$.

$$\text{If } \varphi \in \mathcal{L}^{+}_{LGD}\text{, then } [C]\varphi \text{ and } [\![C]\!]\varphi \text{ are formulas of } \mathcal{L}^{+}_{LGD}.$$

Formula $[C]\varphi$ says that coalition C has a joint strategy to achieve φ at the next state. Formula $[\![C]\!]\varphi$ says that coalition C has a joint strategy for maintaining φ from now on. As we will see from the semantics, these operators are counterparts of $\langle\!\langle C \rangle\!\rangle \bigcirc$ and $\langle\!\langle C \rangle\!\rangle \Box$ in ATL [Alur et al., 2002].

With the full expressiveness of the language, besides game descriptions and strategy representation, we can also use it to describe and compare properties of a game strategy. Let us go back to Example 2.1: mk-games introduced in Chapter 2. Consider the following formulas:

1. $[\![\mathsf{x}]\!](terminal \to wins(\mathsf{x}))$

2. $[\![\mathsf{x}]\!](terminal \to \neg wins(\mathsf{o}))$

3. $[\![\mathsf{x}]\!](terminal \to wins(\mathsf{x})) \triangledown [\![\mathsf{x}]\!](terminal \to \neg wins(\mathsf{o}))$

Formula 1 says player x has a winning strategy and Formula 2 states player x has a no-losing strategy. The last formula expresses that a winning strategy is more preferable than a no-losing strategy for player x.

In order to provide the interpretations for the coalition operators with respect to the state transition model, we need some additional notations. Recall that in Section 2.2.1 a strategy for an agent r is defined as a total function $f_r : W^{\mathcal{P}} \backslash T \to A^r$ such that for all $W^{\mathcal{P}} \backslash T$, $(w, f_r(w)) \in L$. Consequently, a joint strategy for a coalition is a combination of its members' strategies.

Definition 3.2. A *joint strategy* for a coalition $C \subseteq N$ is a total function $f_C : W^{\mathcal{P}} \backslash T \to \Pi_{r \in C} A^r$ such that $f_C(w) = \langle f_r(w) \rangle_{r \in C}$.

The set of all joint strategies of coalition C is denoted by F_C. We say that a complete path δ *complies with* agent r's strategy f_r, if for all $w \in W^{\delta} \backslash T$, for all $j \in \mathbb{N}$, $\delta[j] = w$ implies $\theta_r(\delta, j) = f_r(w)$. That is, for any reachable state w on δ, the action of agent r taken at w on δ is the same as what strategy f_r specifies. Similarly, a complete path δ *complies with* a joint strategy $f_C = \langle f_r(w) \rangle_{r \in C}$, if for all $w \in W^{\delta} \backslash T$, for all $j \in \mathbb{N}$ and for any $r \in C$, $\delta[j] = w$ implies $\theta_r(\delta, j) = f_r(w)$. That is, for any reachable state w on δ, each member r in coalition C takes the same action as what her own strategy f_r specifies at w.

During game playing, a player often begins to adopt a strategy from now on. Accordingly, a complete path δ may start to comply with a joint strategy f_C from the current state. To make this idea precise, let $\mathcal{P}(f_C, w)$ denote the set of complete paths δ reaching state w at some stage j where coalition C starts to use joint strategy f_C. Formally,

$$\mathcal{P}(f_C, w) = \{\delta \in \mathcal{P} \mid \exists j \ \delta[j] = w \text{ and } \forall r \in C, \forall j \leq i \leq |\delta| \ \theta_r(\delta, i) = f_r(\delta[i])\}$$

We are now in the position to provide truth conditions for the coalition operators based on Definition 2.5 as follows:

- $M, \delta, j \models [C]\varphi$ iff $\exists f_C \in F_C \ \forall \delta' \in \mathcal{P}(f_C, \delta[j]) \ \forall j' \in \mathbb{N}$,
  if $\delta[j] = \delta'[j']$ and $j' < |\delta'|$, then $M, \delta', j' + 1 \models \varphi$.
- $M, \delta, j \models [\![C]\!]\varphi$ iff $\exists f_C \in F_C \ \forall \delta' \in \mathcal{P}(f_C, \delta[j]) \ \forall j' \in \mathbb{N}$,
  if $\delta[j] = \delta'[j']$, then $M, \delta', i \models \varphi$ for $\forall i \geq j'$.

The interpretations for the coalition operators are similar to these in ATL [Alur et al., 2002]. Formula $[C]\varphi$ (or $[\![C]\!]\varphi$) is true if coalition C has a joint strategy to make φ true in the next stage (or maintain φ from now on) for all possible complete paths complying with joint strategy f_C starting from state $\delta[j]$. Note that index j' denotes the stage when a complete path δ' reaches state $\delta[j]$. It is possible that $j \neq j'$, since two complete paths may reach the same state at different stages.

By putting all things together, we call the resulting logical framework *the logic for Game Description and strategic Reasoning*, denoted as GDR. We use $\mathcal{L}_{GDR}$ to denote the full language generated by the formation rules given by Definition 2.4 including the formation rules that allows to use of prioritized connectives as well as coalition operators. Formally,

Definition 3.3. A formula φ in $\mathcal{L}_{GDR}$ is defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid initial \mid terminal \mid legal(a^r) \mid$$
$$wins(r) \mid does(a^r) \mid \bigcirc\varphi \mid \varphi \triangledown \varphi \mid [C]\varphi \mid [\![C]\!]\varphi$$

where $p \in \Phi$, $r \in N$, $a^r \in A^r$ and $C \subseteq N$.

The semantics of the language is defined as follows:

Definition 3.4. Let $M = (W, \overline{w}, T, L, U, g, \pi)$ be an ST-model. Given a complete path δ of M, a stage j of δ and a formula $\varphi \in \mathcal{L}_{GDR}$, we say φ is true (or satisfied) at j of δ under M, denoted by $M, \delta, j \models \varphi$, according to the following definition:

$M, \delta, j \models p$	iff	$p \in \pi(\delta[j])$
$M, \delta, j \models \neg\varphi$	iff	$M, \delta, j \not\models \varphi$
$M, \delta, j \models \varphi_1 \wedge \varphi_2$	iff	$M, \delta, j \models \varphi_1$ and $M, \delta, j \models \varphi_2$
$M, \delta, j \models does(a^r)$	iff	$\theta_r(\delta, j) = a^r$
$M, \delta, j \models legal(a^r)$	iff	$(\delta[j], a^r) \in L$
$M, \delta, j \models initial$	iff	$\delta[j] = \overline{w}$
$M, \delta, j \models terminal$	iff	$\delta[j] \in T$
$M, \delta, j \models wins(r)$	iff	$\delta[j] \in g(r)$
$M, \delta, j \models \bigcirc\varphi$	iff	if $j < \lvert\delta\rvert$, then $M, \delta, j+1 \models \varphi$
$M, \delta, j \models \varphi_1 \triangledown \varphi_2$	iff	$M, \delta, j \models \varphi_1$, or $(\mathcal{P}(\varphi_1, \delta[0, j]) = \emptyset$ and $M, \delta, j \models \varphi_2)$
$M, \delta, j \models [C]\varphi$	iff	$\exists f_C \in F_C\ \forall \delta' \in \mathcal{P}(f_C, \delta[j])\ \forall j' \in \mathbb{N}$ if $\delta[j] = \delta'[j']$ and $j' < \lvert\delta'\rvert$, then $M, \delta', j'+1 \models \varphi$.
$M, \delta, j \models [\![C]\!]\varphi$	iff	$\exists f_C \in F_C\ \forall \delta' \in \mathcal{P}(f_C, \delta[j])\ \forall j' \in \mathbb{N}$ if $\delta[j] = \delta'[j']$, then $M, \delta', i \models \varphi$ for $\forall i \geq j'$.

We end this section by the following proposition which says that the coalitional abilities specified by coalition operators satisfy the properties of effectivity functions, except the C-Maximality in Section 3.1.1.

Proposition 3.1. *For all* C, $C' \subseteq N$ *and all* $\varphi, \psi \in \mathcal{L}_{GDR}$,

1. $\models \neg terminal \to \neg[C]\bot$ *(that is, no coalition* C *can enforce the falsity.)*

2. $\models [C]\top$ *(that is, every coalition C can enforce the truth.)*

3. $\models [C](\varphi \wedge \psi) \to [C]\varphi$ *(the outcome-monotonicity)*

4. $\models [C]\varphi \to [C']\varphi$ *where* $C \subseteq C'$ *(the coalition-monotonicity)*

5. $\models [C]\varphi \wedge [C']\psi \to [C \cup C'](\varphi \wedge \psi)$ *where* $C \cap C' = \emptyset$ *(the superadditivity)*

6. $\models \neg terminal \wedge [C]\varphi \to \neg[N \backslash C]\neg\varphi$ *(the C-regularity)*

Similarly for $[\![C]\!]$ *operator.*

Proof. Let M be an arbitrary ST-model, δ be a complete path in M and j be a stage on δ.

1. Suppose $M, \delta, j \models \neg terminal$, then $\delta[j] \notin T$, so for all $f_C \in F_C$, for any $\delta' \in \mathcal{P}(f_C, \delta[j])$ and for any $j' \in \mathbb{N}$ with $\delta[j] = \delta'[j']$ and $j' < |\delta'|$, $M, \delta', j'+1 \not\models \bot$. Thus, $M, \delta, j \not\models [C]\bot$, i.e., $M, \delta, j \models \neg[C]\bot$.

2 and 3 are straightforward.

4. Assume $M, \delta, j \models [C]\varphi$ and $C \subseteq C'$, then there is $f_C \in F_C$, for any $\delta' \in \mathcal{P}(f_C, \delta[j])$ and for any $j' \in \mathbb{N}$ with $\delta[j] = \delta'[j']$ and $j' < |\delta'|$, $M, \delta', j'+1 \models \varphi$. Let $f_{C'}$ be the same as f_C for any $r \in C$. Then it is easy to check that $f_{C'}$ is the joint strategy for coalition C' to achieve φ at the next state. So $M, \delta, j \models [C']\varphi$.

5. Assume $M, \delta, j \models [C]\varphi \wedge [C']\psi$ and $C \cap C' = \emptyset$, then there is $f_C = \langle f_r^1 \rangle_{r \in C}$ for coalition C to achieve φ at the next state, and there is $f_{C'} = \langle f_r^2 \rangle_{r \in C'}$ for coalition C' to achieve ψ at the next state. We define the joint strategy $f_{C \cup C'} = \langle f_r \rangle_{r \in C \cup C'}$ as follows: for any state $w \in \mathcal{P}^w \backslash T$ and any $r \in C \cup C'$,

$$f_r(w) = \begin{cases} f_r^1(w) & \text{if } r \in C; \\ f_r^2(w) & \text{if } r \in C'. \end{cases}$$

This is well-defined as $C \cap C' = \emptyset$. It is easy to check that $f_{C \cup C'}$ is the joint strategy for coalition $C \cup C'$ to achieve both φ and ψ at the next state. Thus, $M, \delta, j \models [C \cup C'](\varphi \wedge \psi)$.

6. Suppose that $\not\models \neg \mathit{terminal} \wedge [C]\varphi \to \neg[N \backslash C]\neg\varphi$, then there is an ST-model M, a complete path δ and a stage j such that $M, \delta, j \not\models \neg \mathit{terminal} \wedge [C]\varphi \to \neg[N \backslash C]\neg\varphi$, then $M, \delta, j \models \neg \mathit{terminal} \wedge [C]\varphi$ and $M, \delta, j \models [N \backslash C]\neg\varphi$. Since $C \cap N \backslash C = \emptyset$, so by 5 we have $M, \delta, j \models \neg \mathit{terminal} \wedge [N]\bot$, contradicting with 1. Thus, $\models \neg \mathit{terminal} \wedge [C]\varphi \to \neg[N \backslash C]\neg\varphi$. □

Regarding the C-Maximality, since not all games are determined, so this property fail to hold in general. Here is a counter-example for the N-Maximality: $\neg[\emptyset]\neg\varphi \to [N]\varphi$. Consider the following initial segment of a complete path δ in a single-player game: $\overline{w} \to \xrightarrow{a} \overline{w} \xrightarrow{b} \cdots$ and formula $does(a) \wedge \bigcirc does(b)$, which says "do action a first and then do action b". It is not hard to check that

$$M, \delta, 0 \not\models \neg[\emptyset]\neg(does(a) \wedge \bigcirc does(b)) \to [1](does(a) \wedge \bigcirc does(b)).$$

We have $M, \delta, 0 \models does(a) \wedge \bigcirc does(b)$, but according to the definition of strategy, it is impossible for player 1 to take different actions at the same state $\overline{w}$, i.e., $M, \delta, 0 \not\models [1](does(a) \wedge \bigcirc does(b))$.

3.2 Strategic Reasoning

In this section, we show how to use the logical framework GDR to reason about game strategies and coalitional abilities. Inherited from GDL, GDR is able to describe the rules of perfect information games. Furthermore, with prioritized connectives and coalition operators, it allows us to represent game strategies and specify coalitional abilities. More importantly, with a unified semantics for GDL- and ATL-formulas, GDR can be used for strategic reasoning.

3.2.1 Game-Oriented Principles

We first formalize Johan Benthem's description of game-oriented principles [van Benthem, 2013], and then use them to derive two well-known results in *combinatorial game theory* [Albert et al., 2007].

The game-oriented principles specify the fundamental properties of any finite turn-taking games of perfect information. These games can be treated as concurrent games, where the players who do not have the turn take the action *noop*. These principles can be represented and proved in GDR as follows:

Theorem 3.2. *Let $\mathcal{F}$ be the ST-frame for the class of finite turn-taking games of perfect information. For $r \in N$ and $\varphi \in \mathcal{L}_{GDR}$, if φ does not contain $\bigcirc$ and $does(.)$, then*

$$\textbf{(A1)}\ \ \mathcal{F} \models [\![r]\!]\varphi \leftrightarrow \varphi \wedge \begin{pmatrix} terminal \vee (turn(r) \wedge [N][\![r]\!]\varphi) \\ \vee \bigvee_{r' \in N \setminus \{r\}} (turn(r') \wedge [\emptyset][\![r]\!]\varphi) \end{pmatrix}.$$

$$\textbf{(A2)}\ \ \mathcal{F} \models \varphi \wedge [\![\emptyset]\!] \begin{pmatrix} (turn(r) \wedge \varphi \rightarrow [N]\varphi) \\ \wedge \bigwedge_{r' \in N \setminus \{r\}} (turn(r') \wedge \varphi \rightarrow [\emptyset]\varphi) \end{pmatrix} \rightarrow [\![r]\!]\varphi.$$

Before proving this theorem, let us first give the intuitions behind the two principles. Principle (A1) is a fixed-point recursion, which says that to maintain a property for a whole game, a player must make sure that it is true now and, before the game is terminated, either there is a strategy so that she can maintain the property at next state if it is her turn, or she can maintain the property at next state for all strategies of other player who has the turn. Note that the property under consideration must be state-wise (no time spanning). Principle (A2) provides a sufficient condition for a player to construct a strategy that maintains a property from now on. A player has a strategy to maintain a property for a whole game if (i) this property holds at the current state; (ii) the player can achieve this property at the next state if it is her turn, and (iii) this property always holds at the next state if it is not her turn. (ii)-(iii) hold recursively. We now prove the theorem. The following lemma will be used for the main proof.

Lemma 3.3. *Given an ST-model M and two complete paths δ, λ in M, let j_1 be a stage on δ and j_2 a stage on λ, if $\delta[j_1] = \lambda[j_2]$, then for all formulas $\varphi \in \mathcal{L}_{GDR}$ without $\bigcirc$ and does(.) operators,*

$$M, \delta, j_1 \models \varphi \textit{ iff } M, \lambda, j_2 \models \varphi.$$

Proof. Given any formula $\varphi \in \mathcal{L}_{GDR}$ without $\bigcirc$ and *does*(.) operators, we next prove this result by induction on the structure of φ except $does(a^r)$ and $\bigcirc\psi$.

- It is straightforward for $p, \neg\psi, \varphi_1 \wedge \varphi_2, initial, terminal, legal(a^r), wins(r)$;
- $\varphi := \varphi_1 \nabla \varphi_2$.

  Assume $M, \delta, j_1 \models \varphi_1 \nabla \varphi_2$, then $M, \delta, j_1 \models \varphi_1$, or $\mathcal{P}(\varphi_1, \delta[0, j_1]) = \emptyset$ and $M, \delta, j_1 \models \varphi_2$. So by Induction Hypothesis we have $M, \lambda, j_2 \models \varphi_1$, or $\mathcal{P}(\varphi_1, \delta[0, j_1]) = \emptyset$ and $M, \lambda, j_2 \models \varphi_2$. And $\delta[j_1] = \lambda[j_2]$, then $\mathcal{P}(\varphi_1, \lambda[0, j_2]) = \emptyset$, otherwise by Lemma 3.3 $\mathcal{P}(\varphi_1, \delta[0, j_1]) \neq \emptyset$, a contradiction. Thus, $M, \lambda, j_2 \models \varphi_1 \nabla \varphi_2$. The other direction is proved in a similar way.
- $\varphi := [C]\psi$.

  Assume $M, \delta, j_1 \models [C]\psi$, then $\exists f_C$ such that $\forall \delta' \in \mathcal{P}(f_C, \delta[j_1])$, $\forall i \in \mathbb{N}$, if $\delta[j_1] = \delta'[i]$, then $M, \delta', i+1 \models \psi$. Let us take the same joint strategy f_C at stage j_2 on complete path λ. Then by Induction Hypothesis and Lemma 3.3, we have $\forall \lambda' \in \mathcal{P}(f_C, \lambda[j_2])$, $\forall l \in \mathbb{N}$, if $\lambda[j_2] = \lambda'[l]$, then $M, \lambda', l+1 \models \psi$. Thus, $M, \lambda, j \models [C]\psi$. The other direction is proved in a similar way.
- $\varphi := [\![C]\!]\psi$. It is proved in a similar way of $[C]\varphi$.

Thus, the result holds. □

In particular, if $\delta[j_1] = \lambda[j_2] \in T$, this result holds for all $\varphi \in \mathcal{L}_{GDR}$. With this lemma, we are now in the position to prove Theorem 3.2.

Proof. (A1) For any ST-model M based on $\mathcal{F}$, any complete path δ in M and any stage j on δ, assume $M, \delta, j \models (terminal \wedge \varphi) \vee (turn(r) \wedge \varphi \wedge [N][\![r]\!]\varphi) \vee \bigvee_{r' \in N \setminus \{r\}}(turn(r') \wedge \varphi \wedge [\emptyset][\![r]\!]\varphi)$. We next prove by cases.

- If $M, \delta, j \models terminal \wedge \varphi$, then by Lemma 3.3, for all $\lambda \in \mathcal{P}(\delta[j])$ with $\lambda[j'] = \delta[j]$, $M, \lambda, j' \models \varphi$, so $M, \delta, j \models [\![r]\!]\varphi$.

- If $M, \delta, j \models turn(r) \wedge \varphi \wedge [N][\![r]\!]\varphi$, then $turn(r) \in \pi(\delta[j])$ and $M, \delta, j \models [N][\![r]\!]\varphi$. By the latter, we have that there is a unique complete path $\lambda \in \mathcal{P}(f_N, \delta[j])$ such that $\forall j' \in \mathbb{N}$ with $\lambda[j'] = \delta[j]$ $M, \lambda, j'+1 \models [\![r]\!]\varphi$. We take the same strategy f_r at stage $j'+1$ on path λ to achieve φ except $f_r(\delta[j]) = \theta_r(\lambda, j')$. Then $\forall \delta' \in \mathcal{P}(f_r, \delta[j])$ $\forall i \in \mathbb{N}$ with $\delta'[i] = \delta[j]$, by $turn(r) \in \pi(\delta[j])$ and $f_r(\delta[j]) = \theta_r(\lambda, j')$, we have $\delta'[i+1] = \lambda[j'+1]$, so $\delta' \in \mathcal{P}(f_r, \lambda[j'+1])$. By the assumption, we obtain $M, \delta', i' \models \varphi$ for all $i' \geq i+1$. And $M, \delta, j \models \varphi$, so by Lemma 3.3, we obtain $M, \delta', i \models \varphi$. Thus, $M, \delta, j \models [\![r]\!]\varphi$.

- If $M, \delta, j \models \bigvee_{r' \in N \setminus \{r\}}(turn(r') \wedge \varphi \wedge [\emptyset][\![r]\!]\varphi)$, then $turn(r') \in \pi(\delta[j])$ for some $r' \in N \setminus \{r\}$, $M, \delta, j \models \varphi$ and $M, \delta, j \models [\emptyset][\![r]\!]\varphi$, so for all $\lambda \in \mathcal{P}(f_\emptyset, \delta[j])$ for all $j' \in \mathbb{N}$ if $\lambda[j'] = \delta[j]$, then $M, \lambda, j'+1 \models [\![r]\!]\varphi$. We take the same strategy f_r at stage $j'+1$ on complete path λ to achieve φ. Then $\forall \delta' \in \mathcal{P}(f_r, \delta[j])$ $\forall i \in \mathbb{N}$ with $\delta'[i] = \delta[j]$, we obtain $M, \delta', i' \models \varphi$ for all $i' \geq i+1$. And $M, \delta, j \models \varphi$, then by Lemma 3.3, we get $M, \delta', i \models \varphi$. Thus, $M, \delta, j \models [\![r]\!]\varphi$.

Therefore, $M, \delta, j \models [\![r]\!]\varphi$.

Conversely, assume $M, \delta, j \models [\![r]\!]\varphi$, then we next prove by two cases: either $\delta[j] \in T$ or $\delta[j] \notin T$.

- If $\delta[j] \in T$, then $M, \delta, j \models terminal$. And by assumption $M, \delta, j \models [\![r]\!]\varphi$ and Lemma 3.3, so $M, \delta, j \models \varphi$.

- If $\delta[j] \notin T$, then either $turn(r) \in \pi(\delta[j])$ or $turn(r') \in \pi(\delta[j])$ for some $r' \in \{N\} \backslash \{r\}$.

  - If $turn(r) \in \pi(\delta[j])$, then $M, \delta, j \models turn(r)$. Suppose not for a contradiction that $M, \delta, j \not\models \varphi$ or $M, \delta, j \not\models [N][\![r]\!]\varphi$.

    * If $M, \delta, j \not\models \varphi$, then by Lemma 3.3, we have that for all $\lambda \in \mathcal{P}(\delta[j])$ and for all $j' \in \mathbb{N}$ with $\lambda[j'] = \delta[j]$, $M, \lambda, j' \not\models \varphi$, so there is no f_r to maintain φ from $\delta[j]$. Thus, $M, \delta, j \not\models [\![r]\!]\varphi$, contradicting with the assumption.

    * If $M, \delta, j \not\models [N][\![r]\!]\varphi$, then for any $\delta_1 \in \mathcal{P}(f_N, \delta[j])$ and for any $j_1 \in \mathbb{N}$ with $\delta_1[j_1] = \delta[j]$ we have $M, \delta_1, j_1 + 1 \not\models [\![r]\!]\varphi$, so for any f_r, there is $\delta_2 \in \mathcal{P}(f_r, \delta_1[j_1 + 1])$ such that for any $j_2 \in \mathbb{N}$ with $\delta_1[j_1 + 1] = \delta_2[j_2]$, there is $l \geq j_2$ such that $M, \delta_2, l \not\models \varphi$. Thus, $M, \delta, j \not\models [\![r]\!]\varphi$, contradicting with the assumption.

    Therefore, $M, \delta, j \models turn(r) \wedge \varphi \wedge [N][\![r]\!]\varphi$.

  - If $turn(r') \in \pi(\delta[j])$, then $M, \delta, j \models turn(r')$. Suppose not for a contradiction that $M, \delta, j \not\models \varphi$ or $M, \delta, j \not\models [\emptyset][\![r]\!]\varphi$.

    * If $M, \delta, j \not\models \varphi$, then by Lemma 3.3, we have that for all $\lambda \in \mathcal{P}(\delta[j])$ and for all $j' \in \mathbb{N}$ with $\lambda[j'] = \delta[j]$, $M, \lambda, j' \not\models \varphi$, so there is no f_r to maintain φ from $\delta[j]$. Thus, $M, \delta, j \not\models [\![r]\!]\varphi$, contradicting with the assumption.

    * If $M, \delta, j \not\models [N][\![r]\!]\varphi$, then for all $\delta_1 \in \mathcal{P}(f_N, \delta[j])$ there is $j_1 \in \mathbb{N}$ such that $\delta_1[j_1] = \delta[j]$ and $M, \delta_1, j_1 + 1 \not\models [\![r]\!]\varphi$, so for any f_r, there is $\delta_2 \in \mathcal{P}(f_r, \delta_1[j_1+1])$ such that for all $j_2 \in \mathbb{N}$ with $\delta_1[j_1+1] = \delta_2[j_2]$ there is $l \geq j_2$ such that $M, \delta_2, l \not\models \varphi$. Let $f_{r'}$ be a strategy for r' such that $f_{r'}(\delta[j]) = \theta_{r'}(\delta_1, j_1)$. Then there is no winning strategy

for r to achieve φ from $\delta[j]$, so $M, \delta, j \not\models [\![r]\!]\varphi$, contradicting with the assumption.

Thus, $M, \delta, j \models \bigvee_{r' \in N \backslash \{r\}} (turn(r') \wedge \varphi \wedge [\emptyset][\![r]\!]\varphi)$.

Thus, $M, \delta, j \models \varphi \wedge (terminal \vee (turn(r) \wedge [N][\![r]\!]\varphi) \vee \bigvee_{r' \in N \backslash \{r\}} (turn(r') \wedge [\emptyset][\![r]\!]\varphi))$.

(A2) For any ST-model M based on $\mathcal{F}$, any complete path δ in M and any stage j on δ, assume

$M, \delta, j \models \varphi \wedge [\![\emptyset]\!]((turn(r) \wedge \varphi \rightarrow [N]\varphi) \wedge \bigwedge_{r' \in N \backslash \{r\}} (turn(r') \wedge \varphi \rightarrow [\emptyset]\varphi))$

iff $M, \delta, j \models \varphi$ and $M, \delta, j \models [\![\emptyset]\!]((turn(r) \wedge \varphi \rightarrow [N]\varphi) \wedge \bigwedge_{r' \in N \backslash \{r\}} (turn(r') \wedge \varphi \rightarrow [\emptyset]\varphi))$

iff $M, \delta, j \models \varphi$, and $\forall \lambda \in \mathcal{P}(\delta[j])$ $\forall j' \in \mathbb{N}$ if $\delta[j] = \lambda[j']$, then $\forall i \geq j', M, \lambda, i \models turn(r) \wedge \varphi \rightarrow [N]\varphi$ and $M, \lambda, i \models \bigwedge_{r' \in N \backslash \{r\}} (turn(r') \wedge \varphi \rightarrow [\emptyset]\varphi)$ $(\star)$

Next we prove by three cases: $\delta[j] \in T$, or $turn(r) \in \pi(\delta[j])$, or $turn(r') \in \pi(\delta[j])$ for some $r' \in N \backslash \{r\}$.

- If $\delta[j] \in T$, then $M, \delta, j \models terminal$. And $M, \delta, j \models \varphi$, so $\forall \lambda \in \mathcal{P}(\delta[j])$ $\forall j' \in \mathbb{N}$ with $\lambda[j'] = \delta[j]$, we have $\lambda[j'] \in T$ and $M, \lambda, j' \models \varphi$ by Lemma 3.3. Thus, $M, \delta, j \models [\![r]\!]\varphi$.
- If $turn(r) \in \pi(\delta[j])$, then $M, \delta, j \models turn(r)$. And by $(\star)$, we have $M, \delta, j \models [N]\varphi$, then $\exists \delta_1 \in \mathcal{P}(\delta[j])$ such that $\forall j_1 \in \mathbb{N}$ with $\delta_1[j_1] = \delta[j]$, $M, \delta_1, j_1 + 1 \models \varphi$. And from $M, \delta, j \models \varphi$ and by Lemma 3.3, we have $M, \delta_1, j_1 \models \varphi$. Let f_r be a strategy of r such that $f_r(\delta[j]) = \theta_r(\delta_1[j_1])$. Then $\delta_1[j_1 + 1] \in T$ or $t(\delta_1[j_1 + 1]) = r$ or $t(\delta_1[j_1 + 1]) = r'$ for some $r' \in N \backslash \{r\}$. Repeat this process until reaching a terminal state where φ holds.
- If $turn(r') \in \pi(\delta[j])$ for some $r' \in N \backslash \{r\}$, then $M, \delta, j \models turn(r')$. And by $(\star)$ and $M, \delta, j \models [\emptyset]\varphi$, so $\forall \delta_2 \in \mathcal{P}(\delta[j])$ $\forall j_2 \in \mathbb{N}$ with $\delta_2[j_2] = \delta[j]$, we have $M, \delta_2, j_2 + 1 \models \varphi$. Then by Lemma 3.3, we obtain $M, \delta_2, j_2 \models \varphi$, so $\delta_2[j_2 + 1] \in T$ or $t(\delta_2[j_2 + 1]) = r$ or $t(\delta_2[j_2 + 1]) = r'$ for some $r' \in N \backslash \{r\}$. Repeat this process until reaching a terminal state where φ holds.

In this way, we construct agent r's strategy f_r step by step, which is a strategy to maintain φ from $\delta[j]$. Thus, $M, \delta, j \models [\![r]\!]\varphi$. □

It is easy to see that these two statements are the generalisation of Benthem's game-oriented principles (Fact 3: C1-C2 in [van Benthem, 2013]). We generalise them into the multi-agent case. Interestingly, the well-known results for finite turn-taking two-player games of perfect information, namely *weak determinacy* and *Zermelo's theorem*, are corollaries of the above theorem.

Proposition 3.4 (Weak Determinacy). *Let $\mathcal{F}^*$ be a state transition frame for the class of finite turn-taking two-player games of perfect information with $N = \{r_1, r_2\}$, then $\mathcal{F}^* \models [\![r_1]\!](terminal \to wins(r_1)) \vee [\![r_2]\!]\neg[\![r_1]\!](terminal \to wins(r_1))$.*

Proof. According to (A1), we have the following valid formulas:

$$\mathcal{F}^* \models turn(r_1) \wedge \varphi \wedge \neg[\![r_1]\!]\varphi \to [\emptyset]\neg[\![r_1]\!]\varphi \tag{3.1}$$

$$\mathcal{F}^* \models turn(r_2) \wedge \varphi \wedge \neg[\![r_1]\!]\varphi \to [N]\neg[\![r_1]\!]\varphi \tag{3.2}$$

We substitute φ with $terminal \to wins(r_1)$, then

$$\mathcal{F}^* \models turn(r_1) \wedge \neg[\![r_1]\!](terminal \to wins(r_1)) \to [\emptyset]\neg[\![r_1]\!](terminal \to wins(r_1)) \tag{3.3}$$

$$\mathcal{F}^* \models turn(r_2) \wedge \neg[\![r_1]\!](terminal \to wins(r_1)) \to [N]\neg[\![r_1]\!](terminal \to wins(r_1)) \tag{3.4}$$

So, we get

$$\mathcal{F}^* \models [\![\emptyset]\!]((turn(r_1)\wedge\neg[\![r_1]\!](terminal \to wins(r_1)) \to [\emptyset]\neg[\![r_1]\!](terminal \to wins(r_1))$$

$$\wedge\, (turn(r_2) \wedge \neg[\![r_1]\!](terminal \to wins(r_1)) \to [N]\neg[\![r_1]\!](terminal \to wins(r_1)))) \tag{3.5}$$

According to (A2), we have

$$\mathcal{F}^* \models \varphi \wedge [\![\emptyset]\!]((turn(r_2) \wedge \varphi \to [N]\varphi) \wedge (turn(r_1) \wedge \varphi \to [\emptyset]\varphi)) \to [\![r_2]\!]\varphi \quad (3.6)$$

We substitute φ with $\neg[\![r_1]\!](terminal \to wins(r_1))$, then

$$\mathcal{F}^* \models \neg[\![r_1]\!](terminal \to wins(r_1)) \wedge [\![\emptyset]\!]((turn(r_2) \wedge \neg[\![r_1]\!](terminal \to wins(r_1))$$

$$\to [N]\neg[\![r_1]\!](terminal \to wins(r_1))) \wedge (turn(r_1) \wedge \neg[\![r_1]\!](terminal \to wins(r_1))$$

$$\to [\emptyset]\neg[\![r_1]\!](terminal \to wins(r_1)))) \to [\![r_2]\!]\neg[\![r_1]\!](terminal \to wins(r_1)) \quad (3.7)$$

Then by 3.5 and 3.7, we get

$$\mathcal{F}^* \models \neg[\![r_1]\!](terminal \to wins(r_1)) \to [\![r_2]\!]\neg[\![r_1]\!](terminal \to wins(r_1)) \quad (3.8)$$

Thus, $\mathcal{F}^* \models [\![r_1]\!](terminal \to wins(r_1)) \vee [\![r_2]\!]\neg[\![r_1]\!](terminal \to wins(r_1))$. □

Weak determinacy says in any finite two-player game with perfect information either one player has a winning strategy or the other player has a strategy that ensures her opponent has no winning strategy. Moreover, if the outcomes of such a game are restricted to three cases: win, lose and tie. Then we obtain Zermelo's Theorem.

Proposition 3.5 (Zermelo's theorem)**.**

$$\mathcal{F}^* \models [\![r_1]\!](terminal \to wins(r_1)) \vee [\![r_2]\!](terminal \to wins(r_2))$$

$$\vee([\![r_1]\!](terminal \to Tie) \wedge [\![r_2]\!](terminal \to Tie))$$

where $Tie =_{def} (\neg wins(r_1) \wedge \neg wins(r_2))$.

Proof. We have the following valid formulas in $\mathcal{F}^*$:

1. $\mathcal{F}^* \models ([\![r_1]\!](terminal \to wins(r_1)) \vee [\![r_1]\!](terminal \to Tie)) \leftrightarrow [\![r_1]\!](terminal \to \neg wins(r_2))$

2. $\mathcal{F}^* \models ([\![r_2]\!](terminal \to wins(r_2)) \vee [\![r_2]\!](terminal \to Tie)) \leftrightarrow [\![r_2]\!](terminal \to \neg wins(r_1))$

3. $\mathcal{F}^* \models [\![r_1]\!](terminal \to wins(r_1)) \to [\![r_1]\!](terminal \to \neg wins(r_2))$

4. $\mathcal{F}^* \models [\![r_2]\!]\neg[\![r_1]\!](terminal \to wins(r_1)) \leftrightarrow [\![r_2]\!](terminal \to \neg wins(r_1))$

The result is directly followed from 1-4 and Proposition 3.4. □

Intuitively, Zermelo's theorem says in any finite two-player game with perfect information and three outcomes (win, lose and tie), if the game cannot end in a draw, then one of the two players has a winning strategy [Polak, 2007].

3.2.2 Reasoning about Strategies: A Case Study

Let us now demonstrate with mk-games how to verify whether a game strategy is a winning or no-losing strategy for an agent, and how to reason about coalitional abilities.

A standard strategy stealing argument from combinatorial game theory shows that there is no winning strategy for the second player in any mk-game. This can be described as follows:

Proposition 3.6. $\mathcal{F}_{mk} \models \neg[\![\mathsf{o}]\!](terminal \to wins(\mathsf{o}))$

Proof. Suppose not for a contradiction that the second player o has a winning strategy S in some mk-game. Let the first player x fill an arbitrary grid to begin with, then she pretends to be o, "stealing" o's strategy S and complying with S. And by assumption, x will win the game. If strategy S specifies her to fill a grid that she has already filled, she should choose to fill a square at random again. This will not interfere with the execution of S, and this strategy is always at least as good as S since having an extra marked grid on the board is never a disadvantage. Thus, the existence of a winning strategy S for o implies the existence of a similarly winning strategy for x: a contradiction since the players cannot both have winning strategies in one game. □

As mk-games are a special family of finite two-player games with perfect information, so by Proposition 3.5, we derive that in mk-games the first player x has a no-losing strategy:

Corollary 3.7. $\mathcal{F}_{mk} \models [\![\mathsf{x}]\!](terminal \to \neg wins(\mathsf{o}))$

Instead of the strategy stealing argument, we next use a constructive approach to illustrate this result by showing that the strategies developed in Section 2.2.3 are actually no-losing or winning strategies for players in 53−game and Tic-Tac-Toe, respectively.

Let us first consider 53-game. We now show that the strategy specified by strategy rule S^r_{53} (2.1) is a winning strategy for player x. To this end, we say that a state transition model M *complies with* player r's strategy f_r specified by a strategy rule S^r, denoted by M_{S^r}, if for all complete paths $\delta \in \mathcal{P}(M_{S^r})$, δ complies with f_r. Formally, $\mathcal{P}(M_{S^r}) = \{\delta \in \mathcal{P}(M) : M, \delta \models \neg terminal \to S^r\}$. With this notion, we can verify that the strategy specified by S^{x}_{53} is a winning strategy for player x in 53-game.

Observation 3.8. For all $\delta \in \mathcal{P}(M_{S^{\mathsf{x}}_{53}})$, $M_{53}, \delta \models terminal \to wins(\mathsf{x})$

Proof. For all $\delta \in \mathcal{P}(M_{S^{\mathsf{x}}_{53}})$, we have $M_{53}, \delta \models turn(\mathsf{x}) \to \widehat{S}^{\mathsf{x}}_{53}$. Then x fills the center ($fill\_center^{\mathsf{x}}$) at the initial state. Whatever o chooses to do, according to S^{x}_{53}, player x can form a two in a row in her third move by filling the smallest empty square next to the center.

- If this two is an open two (i.e., with both end open), then player x guarantees to win in her next move by taking action *check*;
- Otherwise, this two can be blocked by o. Such situation only occurs when o fills the square above the center, i.e., $c_{3,4}$ at her first turn. Then x fills the smallest empty square below the center, i.e., $c_{3,2}$ in her second turn which forms a two with one end open. Then in o's second turn he has no choice but to block the threat by filling the open end. Then according to $\widehat{S}^{\mathsf{x}}_{53}$, x fills the smallest empty square next to $c_{3,2}$, i.e., $c_{2,2}$ at her third turn, which forms an open two. Then in her forth turn, x wins by taking action *check* no matter what o does.

Thus, the strategy specified by S^{x}_{53} is a winning strategy for player x. □

It follows that there is a winning strategy for player x in 53-game.

Corollary 3.9. $M_{53} \models [\![\mathsf{x}]\!](terminal \rightarrow wins(\mathsf{x}))$

We next show Tic-tac-toe game can be forced in a draw by proving the strategy specified by strategy rule S^{x}_{33} (2.2) and the strategy specified by strategy rule S^{o}_{33} (2.3) are no-losing strategies for player x and player o, respectively. The key idea for proving a no-losing strategy for a player is to ensure that once the player adopts the strategy, there is no possibility for her opponent to have two simultaneous check actions, called *a double threat*.

First of all, it is not hard to check that once player x adopts the strategy specified by S^{x}_{33}, her opponent o has no possibility to create a double threat. Thus, we have the following observation saying S^{x}_{33} specifies a no-losing strategy for player x in Tic-Tac-Toe.

Observation 3.10. For all $\delta \in \mathcal{P}(M_{S^{\mathsf{x}}_{33}})$, $M_{33}, \delta \models terminal \rightarrow \neg wins(\mathsf{o})$

On the other hand, the following observation shows that the strategy specified by S^{o}_{33} is a no-losing strategy for player o.

Observation 3.11. For all $\delta \in \mathcal{P}(M_{S^{\mathsf{o}}_{33}})$, $M_{33}, \delta \models terminal \to \neg wins(\mathsf{x})$

Proof. For all $\delta \in \mathcal{P}(M_{S^{\mathsf{o}}_{33}})$, we have $M_{33}, \delta \models \neg terminal \to S^{\mathsf{o}}_{33}$. Then we next prove by three cases: at the initial state player x fills a corner, or fills the center, or fills an edge.

- Case 1. If player x fills a corner at the initial state, then $M_{33}, \delta \models initial \to fill\_corner^{\mathsf{x}}$, then when it is player o's turn, he fills the center as his first move.
  - If x fills the opposite corner, then according to S^{o}_{33}, o fills the smallest edge, then x has to block o, which in turn makes o to block x. Afterwards, there is no way for player x to create a double threat, so x cannot win;
  - If x does not fill the opposite corner, then according to S^{o}_{33}, o fills it if there is no threat to him, which directly causes that x cannot create a double threat, which implies that x cannot win.
- Case 2. If player x fills the center at the initial state, then o fills the smallest corner at his first turn. Then x either fills the opposite corner or not.
  - If x fills the opposite corner at her second turn, then o fills the smallest corner at his second turn leading to a threat to player x, then x has to block it, which in turn leads to a threat to o, so o has to block it. Afterwards, there is no possibility for x to win.
  - If x does not fill the opposite corner at her second turn, then two players have to block each other until there is no possibility for player x to create a double threat.
- Case 3. If player x fills an edge at the initial state, then o fills the center at his first turn. Then x either or not fills a square next to the filled edge.

- If x fills a square which is not next to the filled edge, then o fills the smallest corner, then there is no possibility for player x to create a double threat.
- Otherwise, if there is a threat to him, block it; if not, then o fills the smallest besieged corner to avoid a double threat. In both cases, x has to block o, then there is no possibility for x to win.

Thus, $M_{33}, \delta \models terminal \to \neg wins(\mathsf{x})$. □

Therefore, it follows from Observation 3.10 and Observation 3.11 that Tic-tac-toe can be forced in a draw.

3.3 Model Checking

Checking whether a game or a strategy has a property specified by a GDR-formula is reduced to the model checking problem. In this section, we investigate this issue.

The *model-checking problem* for GDR is the following: for a given GDR formula φ, a state transition model M, a complete path δ and a stage j in M, determine whether or not $M, \delta, j \models \varphi$. By establishing a translation from GDR to ATL*, we show an upper bound of the model-checking problem for GDR, which is not worse than the model-checking problem for ATL*.

Let us first give a brief review of the logic ATL* [Alur et al., 2002]. There are two types of formulas in ATL*: *state formulas* and *path formulas*. They are simultaneously defined as follows:

Let Φ be a finite set of atomic propositions and $N = \{r_1, \cdots, r_k\}$ be a nonempty finite set of agents. The language of ATL*, denoted by $\mathcal{L}_{ATL^*}$, is defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle C \rangle\rangle \psi$$

$$\psi \ ::= \ \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \bigcirc\psi \mid \psi\mathcal{U}\psi$$

where $p \in \Phi$ and $C \subseteq N$.

The semantics of ATL* is based on the *action-based alternating transition system* (AATS) [van der Hoek et al., 2005]. An AATS $\mathcal{T}$ is a tuple $(\Phi, N, W, \overline{w}, A_{r_1}, \cdots, A_{r_k}, \rho, \tau, \pi)$ where

- Φ is a nonempty finite set of *atomic propositions*;
- $N = \{r_1, \cdots, r_k\}$ be a nonempty finite set of *agents*;
- W ia a nonempty set of *states*;
- $\overline{w} \in W$ is the *initial state*;
- A_r is a nonempty finite set of *actions* for each agent $r \in N$, where $A_{r_i} \cap A_{r_j} = \emptyset$ for all $r_i \neq r_j \in N$;
- $\rho : \bigcup_{r \in N} A_r \to 2^W$ is an *action precondition function*, which for any action $a \in \bigcup_{r \in N} A_r$ specifies the set of states $\rho(a)$ from which a may be executed;
- $\tau : W \times \prod_{r \in N} A_r \hookrightarrow W$ is a partial *system transition function*, which for $w \in W$ and joint action $d \in \prod_{r \in N} A_r$ defines the state $\tau(w, d)$ that would result by the performance of d from state w;
- $\pi : \Phi \to 2^W$ is a standard *valuation function*.

Given $d \in \prod_{r \in N} A_r$, let $d(r)$ denote the individual action of agent r in d. A *computation* λ is an infinite sequence of states and actions $\overline{w} \xrightarrow{d_1} w_1 \xrightarrow{d_2} w_2 \cdots$, where for each $j \geq 1$ and $r \in N$, $w_{j-1} \in \rho(d_j(r))$ and $\tau(w_{j-1}, d_j) = w_j$. A computation λ starting in state w is referred to as a *q-computation*. We use $\lambda[j]$ to denote the j-th state on computation λ, and $\lambda[j, \infty]$ to denote the infinite suffix of λ starting from j.

A *strategy* f_r for an agent $r \in N$ is a total function $f_r : W \to A_r$ mapping every state to an action of agent r such that for any $w \in W$, $w \in \rho(f_r(w))$. A *joint strategy* for coalition $C \subseteq N$, denoted by F_C, is a vector of its members' individual strategies, i.e., $\langle f_r \rangle_{r \in C}$. The function $out(w, f_r)$ returns the set of all possible computations that may occur when agent r's strategy f_r executes, starting from state $w \in W$. Formally, $\lambda \in out(w, f_r)$ iff $\lambda[0] = w$ and for any $j \geq 0$, $f_r(\lambda[j]) = \theta_r(\lambda, j)$ where $\theta_r(\lambda, j)$ is the action of agent r taken at stage j on computation λ. The set of all computations complying with joint strategy F_C from state w is defined as $out(w, F_C) = \bigcap_{r \in C} out(w, f_r)$.

The semantics of ATL* can now be given as follows: for an AATS $\mathcal{T}$, a state $w \in W$ and a computation λ of $\mathcal{T}$, the satisfiability relation $\models$ for w and λ of $\mathcal{T}$ is defined inductively as follows:

$\mathcal{T}, w \models p$	iff	$p \in \pi(w)$
$\mathcal{T}, w \models \neg\varphi$	iff	$\mathcal{T}, w \not\models \varphi$
$\mathcal{T}, w \models \varphi_1 \wedge \varphi_2$	iff	$\mathcal{T}, w \models \varphi_1$ and $\mathcal{T}, w \models \varphi_2$
$\mathcal{T}, w \models \langle\langle C \rangle\rangle \psi$	iff	$\exists F_C\ \forall \lambda \in out(w, F_C)\ \mathcal{T}, \lambda \models \psi$
$\mathcal{T}, \lambda \models \varphi$	iff	$\mathcal{T}, \lambda[0] \models \varphi$, where φ is a state formula
$\mathcal{T}, \lambda \models \neg\psi$	iff	$\mathcal{T}, \lambda \not\models \psi$
$\mathcal{T}, \lambda \models \psi_1 \wedge \psi_2$	iff	$\mathcal{T}, \lambda \models \psi_1$ and $\mathcal{T}, \lambda \models \psi_2$
$\mathcal{T}, \lambda \models \bigcirc \psi$	iff	$\mathcal{T}, \lambda[1, \infty] \models \psi$
$\mathcal{T}, \lambda \models \psi_1 \mathcal{U} \psi_2$	iff	$\exists j \geq 0\ \mathcal{T}, \lambda[j, \infty] \models \psi_2$ and $\forall 0 \leq i < j\ \lambda[i, \infty] \models \psi_1$

It has been proved that *the model-checking problem for ATL* with respect to above semantics is PSPACE-complete* [Schobbens, 2004].

3.3.1 From GDR Model to ATL* Model

Let us first show that any state transition model can be transformed into a ATL* model, using the methods introduced by Ruan et al. [2009]. The main idea is that we encode notions like *terminal, legal, wins* through valuation π, rather than through separate relations or functions. For this purpose, we redefine the set of *atomic propositions* of GDR, denoted by At_{GDR}, as follows:

$$At_{\text{GDR}} =_{def} \Phi \cup \{terminal\} \cup \{legal(a^r)) \mid a^r \in A^r\} \cup \{wins(r) \mid r \in N\}$$

Given a state transition model $M = (W, \overline{w}, T, L, U, g, \pi)$ with a game signature $\mathcal{S} = (N, \mathcal{A}, At_{\text{GDR}})$, we define an associated AATS $\mathcal{T}_M = (\Phi', N, W', \overline{w}, A'_{r_1}, \cdots, A'_{r_k}, \rho, \tau, \pi')$ with the same set of agents N and the same initial state $\overline{w}$ such that Φ' is constructed in the following manner.

- for all $\tilde{p} \in At_{\text{GDR}^*}$, $\tilde{p} \in \Phi'$.
- $done(a^r) \in \Phi'$ for all $a^r \in A'_r$ representing actions that are done in the transition from the previous state to the current state[1].
- $initial \in \Phi'$ and $s_\perp \in \Phi'$ where $s_\perp$ is a special atom to specify a 'sink state' which is the only successor of a terminal state and itself.

The other components of $\mathcal{T}_M$ are constructed as follows:

- $W' = W_1 \cup W_2$ where $W_1 = W$ and $W_2 = \{s_w \mid w \in W_1\}$ including sink states.
- $A'_r = A^r \cup \{fin_r\}$ where for all $r \in N$ A^r is in M and fin_r is an action for the terminal and sink states.
- $\rho : \bigcup_{r \in N} A'_r \mapsto W'$ is an action precondition function such that

[1]Note this concept as well as the following two concepts (sink state and fin_r) is borrowed from [Ruan et al., 2009].

- for any $a \in \mathcal{A}$, $\rho(a) = \{w \in W \mid (w, a) \in L\}$;
- for any $r \in N$, $\rho(fin_r) = W_2 \cup T$.

- $\tau : A'_{r_1} \times \cdots \times A'_{r_k} \times W' \to W'$ is a system transition function such that
  - for all $d \in \Pi_{r \in N} A^r$ and $w \in W$, $\tau(d, w) = U(d, w)$;
  - for all $w \in T$, $\tau(\langle fin_{r_1}, \cdots, fin_{r_k}\rangle, w) = s_w$;
  - for all $s_w \in W_2$, $\tau(\langle fin_{r_1}, \cdots, fin_{r_k}\rangle, s_w) = s_w$.

- $\pi' : W' \mapsto 2^{\Phi'}$ is a valuation function such that for any $w \in W'$ $\pi'(w)$ is a set of atoms satisfying the following conditions:

  (1) for all $w \in W_1$,
  - for any $p \in \Phi$, $p \in \pi(w)$ iff $p \in \pi'(w)$;
  - $w \in T$ iff $terminal \in \pi'(w)$;
  - for any $r \in N$, $w \in g(r)$ iff $wins(r) \in \pi'(w)$;
  - for any $a^r \in \mathcal{A}$, $(w, a^r) \in L$ iff $legal(a^r) \in \pi'(w)$;
  - $initial \in \pi'(\overline{w})$;
  - $done(a^r) \in \pi'(w)$ iff $\theta_r(\delta, j) = a^r$ and $\delta[j+1] = w$[2].

  (2) for all $s_w \in W_2$,
  $\pi'(s_w) = \pi'(w) \cup \{s_\perp\} \cup \{done(fin_r) \mid r \in N\} \backslash \{done(a^r) \mid a^r \in \mathcal{A}\}$.

Given a complete path δ in M, we extend δ to a computation in $\mathcal{T}_M$ with the sink state labelled by the terminal state of δ. We denote it by $\tilde{\delta}$. In particular, we use $\tilde{\delta}[j, \infty]$ to denote the infinite subsequence of $\tilde{\delta}$ starting at stage j.

3.3.2 From GDR Descriptions to ATL* Specifications

We next define a translation map from GDR formulas to ATL* formulas to embed GDR into ATL*.

[2]The idea is that there is a unique action for each player to take at every state and the update is deterministic, so formula does(.) can be translated to a state formula done(.) at the next state.

Definition 3.5. A translation Tr^* from GDR formulas to ATL* formulas is defined as follows:

- $\mathrm{Tr}^*(\tilde{p}) = \mathrm{Tr}(\tilde{p})$ for all $\tilde{p} \in At_{\mathrm{GDR}}$
- $\mathrm{Tr}^*(initial) = initial$
- $\mathrm{Tr}^*(\neg\varphi) = \neg\mathrm{Tr}^*(\varphi)$
- $\mathrm{Tr}^*(\varphi_1 \wedge \varphi_2) = \mathrm{Tr}^*(\varphi_1) \wedge \mathrm{Tr}^*(\varphi_2)$
- $\mathrm{Tr}^*(does(a^r)) = \bigcirc done(a^r)$
- $\mathrm{Tr}^*(\varphi_1 \triangledown \varphi_2) = \mathrm{Tr}^*(\varphi_1) \vee (\langle\langle\emptyset\rangle\rangle\neg\mathrm{Tr}^*(\varphi_1) \wedge \mathrm{Tr}^*(\varphi_2))$
- $\mathrm{Tr}^*(\bigcirc\varphi) = \bigcirc(\neg s_\perp \to \mathrm{Tr}^*(\varphi))$
- $\mathrm{Tr}^*([C]\varphi) = \langle\langle C\rangle\rangle \bigcirc (\neg s_\perp \to \mathrm{Tr}^*(\varphi))$
- $\mathrm{Tr}^*([\![C]\!]\varphi) = \langle\langle C\rangle\rangle\Box\mathrm{Tr}^*(\varphi)$

Then we have the following corresponding result between the state transition model and its associated AATS with respect to the translation.

Lemma 3.12. *Given a state transition model M, a complete path δ in M, a stage j on δ and any formula $\varphi \in \mathcal{L}_{GDR}$,*

(1) if $Tr^*(\varphi)$ is an ATL state formula, $M, \delta, j \models_{GDR} \varphi$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models_{ATL^*} Tr^*(\varphi)$;*

(2) if $Tr^*(\varphi)$ is an ATL path formula, $M, \delta, j \models_{GDR} \varphi$ iff $\mathcal{T}_M, \tilde{\delta}[j, \infty] \models_{ATL^*} Tr^*(\varphi)$.*

Proof. Given a state transition model M, a complete path δ in M, a stage j on δ and for any GDR formulas $\varphi \in \mathcal{L}_{GDR}$, we prove the two statements by induction on the structure of φ simultaneously.

- $\varphi := p$. Assume $M, \delta, j \models p$ iff $p \in \pi(\delta[j])$ iff $p \in \pi'(\delta[j])$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models p$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(p)$.

- $\varphi := initial$. Assume $M, \delta, j \models initial$ iff $\delta[j] = \overline{w}$ iff $initial \in \pi'(\delta[j])$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models initial$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(initial)$.

- $\varphi := terminal$. Assume $M, \delta, j \models terminal$ iff $\delta[j] \in T$ iff $terminal \in \pi'(\delta[j])$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models terminal$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(terminal)$.

- $\varphi := legal(a^r)$. Assume $M, \delta, j \models legal(a^r)$ iff $(\delta[j], a^r) \in L$ iff $legal(a^r) \in \pi'(\delta[j])$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models legal(a^r)$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(legal(a^r))$.

- $\varphi := wins(r)$. Assume $M, \delta, j \models wins(r)$ iff $\delta[j] \in g(r)$ iff $wins(r) \in \pi'(\delta[j])$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models wins(r)$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(wins(r))$.

- $\varphi := does(a^r)$. Assume $M, \delta, j \models does(a^r)$ iff $\theta_r(\delta, j) = a^r$ iff $done(a^r) \in \pi(\delta[j+1])$ iff $\mathcal{T}_M, \tilde{\delta}[j+1] \models done(a^r)$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models \bigcirc done(a^r)$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(does(a^r))$.

- $\varphi := \neg\psi$.
  - If $\mathrm{Tr}^*(\neg\psi)$ is an ATL*-state formula, then $M, \delta, j \models \neg\psi$ iff $M, \delta, j \not\models \psi$ iff $\mathcal{T}_M, \tilde{\delta}[j] \not\models \mathrm{Tr}^*(\psi)$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(\neg\psi)$;
  - If $\mathrm{Tr}^*(\neg\psi)$ is an ATL*-path formula, then $M, \delta, j \models \neg\psi$ iff $M, \delta, j \not\models \psi$ iff $\mathcal{T}_M, \tilde{\delta}[j, \infty] \not\models \mathrm{Tr}^*(\psi)$ iff $\mathcal{T}_M, \tilde{\delta}[j, \infty] \models \mathrm{Tr}^*(\neg\psi)$.

- $\varphi := \varphi_1 \wedge \varphi_2$.
  - If $\mathrm{Tr}^*(\varphi_1 \wedge \varphi_2)$ is an ATL*-state formula, then $M, \delta, j \models \varphi_1 \wedge \varphi_2$ iff $M, \delta, j \models \varphi_1$ and $M, \delta, j \models \varphi_2$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(\varphi_1)$ and $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(\varphi_2)$ iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(\varphi_1 \wedge \varphi_2)$;
  - If $\mathrm{Tr}^*(\varphi_1 \wedge \varphi_2)$ is an ATL*-path formula, then $M, \delta, j \models \varphi_1 \wedge \varphi_2$ iff $M, \delta, j \models \varphi_1$ and $M, \delta, j \models \varphi_2$ iff $\mathcal{T}_M, \tilde{\delta}[j, \infty] \models \mathrm{Tr}^*(\varphi_1)$ and $\mathcal{T}_M, \tilde{\delta}[j, \infty] \models \mathrm{Tr}^*(\varphi_2)$ iff $\mathcal{T}_M, \tilde{\delta}[j, \infty] \models \mathrm{Tr}^*(\varphi_1 \wedge \varphi_2)$.

- $\varphi := \bigcirc\psi$. Assume $M, \delta, j \models \bigcirc\psi$ iff $M, \delta, j+1 \models \psi$ iff $\mathcal{T}_M, \tilde{\delta}[j+1, \infty] \models \neg s_\perp \rightarrow \mathrm{Tr}^*(\psi)$ iff $\mathcal{T}_M, \tilde{\delta}[j, \infty] \models \bigcirc(\neg s_\perp \rightarrow \mathrm{Tr}^*(\psi))$ iff $\mathcal{T}_M, \tilde{\delta}[j, \infty] \models \mathrm{Tr}^*(\bigcirc\psi)$.

- $\varphi := \varphi_1 \nabla \varphi_2$.

  Assume $M, \delta, j \models \varphi_1 \nabla \varphi_2$

  iff $M, \delta, j \models \varphi_1$, or $\mathcal{P}(\varphi_1, \delta[0, j]) = \emptyset$ and $M, \delta, j \models \varphi_2$

  iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(\varphi_1)$, or $\mathcal{P}(\varphi_1, \delta[0, j]) = \emptyset$ and $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(\varphi_2)$

  iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(\varphi_1)$, or for all $\lambda \in \mathit{out}(\delta[j])$ $\mathcal{T}_M, \lambda[0] \not\models \mathrm{Tr}^*(\varphi_1)$ and $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(\varphi_2)$

  iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(\varphi_1) \vee (\langle\langle\emptyset\rangle\rangle \neg \mathrm{Tr}^*(\varphi_1) \wedge \mathrm{Tr}^*(\varphi_2))$

  iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*(\varphi_1 \nabla \varphi_2)$.

- $\varphi := [C]\psi$.

  Assume $M, \delta, j \models [C]\psi$

  iff $\exists f_C$ such that for all $\delta' \in \mathcal{P}(f_C, \delta[j])$ and for all $j' \in \mathbb{N}$, if $\delta'[j'] = \delta[j]$ and $j' < |\delta'|$ then $M, \delta', j'+1 \models \psi$

  iff $\exists f_C$ such that for all $\lambda \in \mathit{out}(f_C, \delta[j'])$, $M, \lambda[1] \models \neg s_\perp \rightarrow \mathrm{Tr}^*(\psi)$

  iff $\mathcal{T}_M, \tilde{\delta}[j] \models \langle\langle C\rangle\rangle \bigcirc (\neg s_\perp \rightarrow \mathrm{Tr}^*(\psi))$

  iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*([C]\psi)$.

- $\varphi := [\![C]\!]\psi$.

  Assume $M, \delta, j \models [\![C]\!]\psi$

  iff $\exists f_C$ such that for all $\delta' \in \mathcal{P}(f_C, \delta[j])$ and for all $j' \in \mathbb{N}$, if $\delta'[j'] = \delta[j]$ then $M, \delta', i \models \psi$ for all $i \geq j'$.

  iff $\exists f_C$ such that for all $\lambda \in \mathit{out}(f_C, \delta[j])$, for all $i \geq 0$, $M, \lambda[i] \models \mathrm{Tr}^*(\psi)$

  iff $\mathcal{T}_M, \tilde{\delta}[j] \models \langle\langle C\rangle\rangle \Box \mathrm{Tr}^*(\psi)$

  iff $\mathcal{T}_M, \tilde{\delta}[j] \models \mathrm{Tr}^*([\![C]\!]\psi)$.

Thus, the result holds. $\square$

And the size of the transformations from GDR to ATL* is polynominal. Thus, the model-checking problem for GDR is not worse than that for ATL*. Since *the model-checking problem for ATL* is PSPACE-complete* [Schobbens, 2004], so we have the following result.

Theorem 3.13. *The model-checking problem for GDR is in PSPACE.*

At last it should be noted that the reduction of GDR to ATL* does not mean that ATL* can be used for strategy representation. It is not hard to find that once we translate the strategies we designed for the *mk*-games into ATL*, we will lose all the intuitions behind these strategies. ATL* is an expressive logical language for specifying strategic abilities of players but certainly not suitable for describing game strategies.

3.4 Summary

Up to now, we have fulfilled our goal by presenting a comprehensive logical framework for reasoning about perfect information games. The language of the framework combines GDL with prioritized connectives for representing game strategies and coalition operators for specifying strategic abilities of game players. To minimize the complexity of this language, we have taken a cautious way of doing that. We did not introduce until operator $\mathcal{U}$ and coalition operators binded with next $\bigcirc$ and always $\Box$. We have demonstrated with a running example that our language is able to describe game rules, express game properties, represent game strategies and specify coalitional abilities. More importantly, the framework allows us to formalize generic game results such as Weak Determinacy and Zermelo's Theorem, verify whether or not a game strategy is winning or no-losing, and reason about coalitional abilities. We have also investigated the model-checking problem for the logic. To the best of our knowledge, there is no other logical system with such computational complexity that can have the same expressive power.

Most of the related work has been discussed in Section 1.2. Besides that, the following is also worth mentioning.

Ruan et al. [2009] investigate the relationship between GDL and ATL. Different from our motivation and method, their goal is to use ATL to reason about GDL-specified games. And instead of integrating language components, they focus on how to transfer a GDL game specification into an ATL specification. Therefore, there is no new logic produced in their paper but using ATL for reasoning about properties of GDL-defined games.

There has been some work to make strategies explicit in ATL by adding a strategy term into coalition operators, meaning that a coalition commits to a strategy [Herzig et al., 2013, van der Hoek et al., 2005, Walther et al., 2007]. In such a way, strategies can be expressed syntactically. However, as pointed out by [Ramanujam and Simon, 2008b, van Eijck, 2013], these strategies are atomic terms in the language level without showing their structures. Thus, it is difficult to use the language to design a strategy so as to achieve a goal state.

Chapter 4

Reasoning about Games with Imperfect Information

Games can be classified as *perfect information games* or *imperfect information games* [Osborne, 1994]. Chapter 2 and Chapter 3 deal with perfect information games. In this chapter, we turn to investigating games with imperfect information, such as Poker, Kriegspiel (chess).

4.1 Background

Playing games with imperfect information poses an intricate reasoning challenge for players, since imperfect information requires a player to use the rules of a game to infer useful game information, draw conclusions from her own knowledge about the current game state and about knowledge of other players. In order to incorporate imperfect information games, GDL has recently been extended to GDL-II [Thielscher, 2010]. However, as a purely descriptive language, GDL-II is only a tool for describing the rules of an imperfect information game, but does not provide a facility for reasoning about how a player infers unveiled information based on the rules [Schiffel and Thielscher, 2011, 2014]. Indeed, some information is essential for players to proceed with a game. For example, players should always know their own available actions in non-terminal states and know their results in terminal states. Such epistemic properties of a game are normally implied by the game rules and thus need reasoning facilities to infer and verify them. Unfortunately, GDL-II (or GDL) is not designed for this purpose.

To handle this issue, a few approaches have been proposed, mostly embedding GDL-II into a logical system, such as Situation Calculus or Alternating-time Temporal Epistemic Logic (ATEL), to use their reasoning facilities [Huang et al., 2013, Ruan and Thielscher, 2011, 2012, Schiffel and Thielscher, 2011]. As long as the targeting logics are expressive enough to interpret any GDL description, it is possible to use the inference mechanisms of these logics for reasoning about GDL-II games. However, a highly expressive logic may incur high complexity for reasoning tasks. For instance, Ruan and Thielscher [2012] propose an adaption of ATEL to verify epistemic properties of GDL-II games, and show that the model-checking problem in that setting is 2EXPTIME-hard. Such high computational complexity may be not what we want.

This chapter aims to propose a different approach to deal with this problem. We introduce a logical framework, called EGDL, equipped with a language for describing imperfect information games and a semantical model that can be used for reasoning about game information and players' epistemic status. Most importantly, we develop a model-checking algorithm for EGDL and show that the complexity of the model-checking problem for the logic can be significantly reduced to Δ_2^p. There are two major reasons that help us to reduce the complexity. Firstly, our language is a conservative extension of GDL with standard epistemic operators [Fagin et al., 2003b]. We take a cautious way of doing that without introducing the until operator or coalition operators. Secondly, we provide an imperfect recall semantics for knowledge. Other cases could be considered; nevertheless, the addition of perfect recall to GDL-II renders the model-checking problem of ATEL undecidable in general [Ruan and Thielscher, 2012]. Also, in many applications, especially when modeling extremely large games, imperfect recall may provide considerably empirical and practical advantages [Busard et al., 2015, Piccione and Rubinstein, 1997, Waugh et al., 2009]. Despite moderate expressive power, we demonstrate with an example that the language is able to express game rules, formalize essential epistemic properties and specify the interactions of knowledge and actions. In this sense, EGDL makes a good balance between expressive power and computational efficiency.

4.2 Epistemic State Transition Structures

Following Chapter 2, in this chapter, all concepts will be based on the same game signature $\mathcal{S} = (N, \mathcal{A}, \Phi)$ defined in Section 2.1. We begin by introducing the semantic structures used to model synchronous games with imperfect information. The underlying structures of these games are specified by epistemic state transition frames which are obtained by adding an epistemic relation for each agent to state transition frames.

Definition 4.1. An *epistemic state transition (ET) frame* $\mathcal{F}$ is a tuple $(W, \overline{w}, T, L, U, g, \{R_r\}_{r\in N})$, where

- W is a nonempty set of *states*.
- $\overline{w} \in W$ is the *initial* state.
- $T \subseteq W$ is the set of *terminal* states.
- $L \subseteq W \times \mathcal{A}$ is a *legality* relation, describing the legal actions at each state.
- $U : W \times D \to W$ is an *update* function, specifying the state transitions for each joint action, where $D = \prod_{r\in N} A^r$ is the set of joint actions.
- $g : N \to 2^W$ is a *goal* function, specifying the winning states for each agent.

- $R_r \subseteq W \times W$ is an equivalence relation for agent r, indicating the states that are indistinguishable for r.

Associating an ET-frame with a standard valuation function, we obtain an epistemic state transition model which can be used to specify a particular game with imperfect information.

Definition 4.2. An *epistemic state transition (ET) model* M is is a pair $(\mathcal{F}, \pi)$ where

- $\mathcal{F}$ is an ET-frame;
- $\pi : W \to 2^{\Phi}$ is a standard valuation function.

Given an ET-model, the definitions of *path* and *complete path* as well as the notations are the same as those in Section 2.1.1.

The following definition, by extending equivalence relations over states to complete paths, characterizes precisely what an agent with imperfect recall and perfect reasoning can in principle know at a specific stage of a game.

Definition 4.3. Two complete paths δ, $\delta' \in \mathcal{P}$ are *imperfect recall* (also called memoryless) equivalent for agent r at stage $j \in \mathbb{N}$, written $\delta \approx_r^j \delta'$, iff $\delta[j] R_r \delta'[j]$.

That is, imperfect recall requires an agent to be only aware of the present state but forget everything that happened. This is similar to the notion of imperfect recall in ATL [Schobbens, 2004]. It should be noted that similar to [Halpern and Vardi, 1989], in synchronous games we assume every agent has access to a global clock, so the agents always know the time (game stage).

To demonstrate the flexibility of the framework, we next present three interesting equivalence relations which specify different memory types of agents.

Definition 4.4. Two complete paths δ, $\delta' \in \mathcal{P}$ are *state-based equivalent* for agent r at stage $j \in \mathbb{N}$, written $(\delta, j) \approx_r^{sr} (\delta', j)$, iff $\delta[l] R_r \delta'[l]$ for any $0 \leq l \leq j$.

Intuitively, this equivalence relation describes agents who remember all the past states of the system, but forget their own actions. This is also called perfect recall in epistemic ATL-style logics [Jamroga and van der Hoek, 2004]. Similarly, we define the equivalence relation for agents who remember all their own actions, but forget the past states, such as agents in a maze [van Ditmarsch and Knight, 2014].

Definition 4.5. Two complete paths δ, $\delta' \in \mathcal{P}$ are *action-based equivalent* for agent r at stage $j \in \mathbb{N}$, written $(\delta, j) \approx_r^{ar} (\delta', j)$, iff $\theta_r(\delta, l) = \theta_r(\delta', l)$ for any $0 \leq l < j$.

It follows that in the limit case $j = 0$, we have that $(\delta, 0) \approx_i^{ar} (\delta', 0)$ always holds. Finally we give the equivalence relation for agents who remember both past states of the system and their own actions. This is similar to the notion of perfect recall in GDL-II [Thielscher, 2010].

Definition 4.6. Two complete paths δ, $\delta' \in \mathcal{P}$ are *perfect recall equivalent* for agent r at stage $j \in \mathbb{N}$, written $(\delta, j) \approx_r^{pr} (\delta', j)$, iff $\delta[l] R_r \delta'[l]$ for any $0 \leq l \leq j$, and $\theta_r(\delta, i) = \theta_r(\delta', i)$ for all $0 \leq i < j$.

The following proposition displays the interrelations of above memory types.

Proposition 4.1. *For all δ, $\delta' \in \mathcal{P}$, any $j \in \mathbb{N}$ and any $r \in N$,*

1. *if $(\delta, j) \approx_r^{sr} (\delta', j)$, then $\delta \approx_r^j \delta'$*
2. *$(\delta, j) \approx_r^{pr} (\delta', j)$ iff $(\delta, j) \approx_r^{sr} (\delta', j)$ and $(\delta, j) \approx_r^{ar} (\delta', j)$.*
3. *if $(\delta, j) \approx_r^x (\delta', j)$, then $(\delta, t) \approx_r^x (\delta', t)$ for any $0 \leq t \leq j$, where $x \in \{sr, ar, pr\}$.*

For practical and computational reasons, we focus on agents with imperfect recall and the other cases are left for future work. To illustrate the framework, let us consider a variant of mk-games in Example 2.1, called Krieg-mk-games. In the rest of this chapter we use it as a running example.

Example 4.1. *A Krieg-mk-game is played by two players, cross* x *and naught* o*, who take turns marking grids on a* $m \times m$ *board. Different from a standard mk-game, each player can see her own marks, but not those of her opponent, just like the chess variant* Kriegspiel *[Pritchard, 1994]. Players are informed of the turn-taking. The game ends if the board is completely filled or one player wins by having completed a horizontal, vertical or diagonal line of* k *with her own symbol.*

Obviously, a Krieg-mk-game is a generalisation of Krieg-Tictactoe ($m = k = 3$) *in [Schiffel and Thielscher, 2011].*

To represent a Krieg-mk-game in terms of the ET-model, we first describe the game signature, written $\mathcal{S}_{KG}$*, as follows:*

- $N_{KG} = \{\mathsf{x}, \mathsf{o}\}$*;*
- $A^r_{KG} = \{a^r_{i,j} \mid 1 \leq i, j \leq m\} \cup \{noop^r\}$*, where* $a^r_{i,j}$ *denotes the action that player* r *fills grid* (i, j) *with her symbol, and* $noop^r$ *denotes that player* r *does action noop;*
- $\Phi_{KG} = \{p^r_{i,j}, tried(a^r_{i,j}), turn(r) \mid r \in \{\mathsf{x}, \mathsf{o}\}$ *and* $1 \leq i, j \leq m\}$*, where* $p^r_{i,j}$ *represents the fact that grid* (i, j) *is filled with player* i*'s symbol,* $tried(a^r_{i,j})$ *represents the fact that player* r *has tried to fill grid* (i, j) *before, and* $turn(r)$ *says that it is player* r*'s turn now.*

Based on this, we specify the ET-frame for this game, written $\mathcal{F}_{KG}$*, as follows:*

- $W_{KG} = \{(t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}) : t_{\mathsf{x}}, t_{\mathsf{o}} \in \{0, 1\}\ \&\ c_{1,1}, \cdots, c_{m,m} \in \{\square, \boxtimes, \boxdot, \otimes, \odot\}\}$ *be the set of possible states, where* t_{x}*,* t_{o} *specify the turn taking and* $c_{i,j}$ *represents the fact that grid* (i, j) *is empty* $\square$*, or occupied by*
  - *the cross and not tried by the nought* $\boxtimes$*,*
  - *the nought and not tried by the cross* $\boxdot$*,*
  - *the nought and tried by the cross* $\otimes$*,*

  - *the cross and tried by the nought* $\odot$.

- $\overline{w}_{KG} = (1, 0, \square, \cdots, \square)$.

- $g_{KG}(\mathsf{x}) = \{(t_\mathsf{x}, t_\mathsf{o}, c_{1,1}, \cdots, c_{m,m}) : c_{i,j}, \cdots, c_{h,l} \in G_k \;\&\; c_{i,j}, \cdots, c_{h,l} \in \{\boxtimes, \odot\}\}$, *and*

  $g_{KG}(\mathsf{o}) = \{(t_\mathsf{x}, t_\mathsf{o}, c_{1,1}, \cdots, c_{m,m}) : c_{i,j}, \cdots, c_{h,l} \in G_k \;\&\; c_{i,j}, \cdots, c_{h,l} \in \{\boxdot, \otimes\}\}$,

  where $1 \leq i, j, h, l \leq m$, *and*

  $$\begin{aligned} G_k = &\{c_{i,j}, \cdots, c_{i,j+k-1} : 1 \leq i \leq m, 1 \leq j \leq m-k+1\} \\ &\cup \{c_{i,j}, \cdots, c_{i+k-1,j} : 1 \leq i \leq m-k+1, 1 \leq j \leq m\} \\ &\cup \{c_{i,j}, \cdots, c_{i+k-1,j+k-1} : 1 \leq i, j \leq m-k+1\} \\ &\cup \{c_{i,j}, \cdots, c_{i+k-1,j-k+1} : 1 \leq i \leq m-k+1, k \leq j \leq m\}^{[1]}. \end{aligned}$$

- $T_{KG} = g_{KG}(\mathsf{x}) \cup g_{KG}(\mathsf{o})$
  $\cup\ \{(t_\mathsf{x}, t_\mathsf{o}, c_{1,1}, \cdots, c_{m,m}) : c_{i,j} \in \{\boxtimes, \boxdot, \otimes, \odot\}$ *for* $1 \leq i, j \leq m\}$.

- *For all states* $w = (t_\mathsf{x}, t_\mathsf{o}, c_{1,1}, \cdots, c_{m,m})$ *and* $w' = (t'_\mathsf{x}, t'_\mathsf{o}, c'_{1,1}, \cdots, c'_{m,m})$ *in* W_{KG},

  - $wR^{KG}_\mathsf{x}w'$ *iff (1)* $t_r = t'_r$ *for any* $r \in N_{KG}$; *(2)* $c_{i,j} \in \{\boxtimes, \odot\}$ *iff* $c'_{i,j} \in \{\boxtimes, \odot\}$ *for any* $1 \leq i, j \leq m$; *(3)* $c_{i,j} = \otimes$ *iff* $c'_{i,j} = \otimes$ *for any* $1 \leq i, j \leq m$.

  - $wR^{KG}_\mathsf{o}w'$ *iff (1)* $t_r = t'_r$ *for any* $r \in N_{KG}$; *(2)* $c_{i,j} \in \{\boxdot, \otimes\}$ *iff* $c'_{i,j} \in \{\boxdot, \otimes\}$ *for any* $1 \leq i, j \leq m$; *(3)* $c_{i,j} = \odot$ *iff* $c'_{i,j} = \odot$ *for any* $1 \leq i, j \leq m$.

- *For all* $(t_\mathsf{x}, t_\mathsf{o}, c_{1,1}, \cdots, c_{m,m}) \in W_{KG}$,

  - *for any* $a^\mathsf{x}_{i,j} \in \mathcal{A}_{KG}$,
    $((t_\mathsf{x}, t_\mathsf{o}, c_{1,1}, \cdots, c_{m,m}), a^\mathsf{x}_{i,j}) \in L_{KG}$ *iff* $t_\mathsf{x} = 1$ *and* $c_{i,j} \in \{\square, \boxdot\}$.

  - *for any* $a^\mathsf{o}_{i,j} \in \mathcal{A}_{KG}$,
    $((t_\mathsf{x}, t_\mathsf{o}, c_{1,1}, \cdots, c_{m,m}), a^\mathsf{o}_{i,j}) \in L_{KG}$ *iff* $t_\mathsf{o} = 1$ *and* $c_{i,j} \in \{\square, \otimes\}$.

[1] G_k is the set of all lines of k grids. For instance, on 3×3 board, G_3 has 8 elements.

- *for* $noop^r \in \mathcal{A}_{KG}$,

  $((t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}), noop^r) \in L_{KG}$ *iff* $t_{-r} = 1$.

- $U_{KG} : W_{KG} \times D_{KG} \to W_{KG}$ *is defined as follows: for all* $(t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}) \in W_{KG}$ *and for all* $\langle a^r_{i,j}, noop^{-r}\rangle \in D_{KG}$, *let*

$$U_{KG}((t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}), \langle a^r_{i,j}, noop^{-r}\rangle) = (t'_{\mathsf{x}}, t'_{\mathsf{o}}, c'_{1,1}, \cdots, c'_{m,m})$$

such that $(t'_{\mathsf{x}}, t'_{\mathsf{o}}, c'_{1,1}, \cdots, c'_{m,m})$ *is the same as* $(t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m})$ *except its components* $t'_{\mathsf{x}}, t'_{\mathsf{o}}$ *and* $c'_{i,j}$ *which are updated as follows:* $t'_{r} = t_{\text{-}r}$ *and*

$$c'_{i,j} = \begin{cases} \boxtimes & \text{if } c_{i,j} = \square \text{ and } r = \mathsf{x}; \\ \boxdot & \text{if } c_{i,j} = \square \text{ and } r = \mathsf{o}; \\ \otimes & \text{if } c_{i,j} = \boxdot \text{ and } r = \mathsf{x}; \\ \odot & \text{if } c_{i,j} = \boxtimes \text{ and } r = \mathsf{o}; \\ c_{i,j} & \text{otherwise.} \end{cases}$$

Finally, for each state $w = (t_{\mathsf{x}}, t_{\mathsf{o}}, c_{1,1}, \cdots, c_{m,m}) \in W_{KG}$, $\pi_{KG}(w) = \{turn(r) : t_r = 1\} \cup \{p^{\mathsf{x}}_{i,j} : c_{i,j} \in \{\boxtimes, \odot\}\} \cup \{p^{\mathsf{o}}_{i,j} : c_{i,j} \in \{\boxdot, \otimes\}\} \cup \{tried(a^{\mathsf{x}}_{i,j}) : c_{i,j} = \otimes\} \cup \{tried(a^{\mathsf{o}}_{i,j}) : c_{i,j} = \odot\}$. *Let* $M_{KG} = (\mathcal{F}_{KG}, \pi_{KG})$ *be the ET-model for this game.*

In addition, we assume that each player takes the same action at the stage of all her indistinguishable complete paths, i.e., for any $\delta, \delta' \in \mathcal{P}(M_{KG})$, $j \in \mathbb{N}$ *and* $r \in N_{KG}$, *if* $\delta \approx^j_r \delta'$, *then* $\theta_r(\delta, j) = \theta_r(\delta', j)$.

4.3 Epistemic Game Description Logic

In this section, we introduce the logical framework EGDL for representing and reasoning about imperfect information games.

4.3.1 The Language

The language, denoted by $\mathcal{L}_{EGDL}$, is obtained by extending GDL with the standard epistemic operators [Fagin et al., 2003b].

Definition 4.7. A formula φ in $\mathcal{L}_{EGDL}$ is defined by the following BNF:

$$\varphi \ ::= \ p \mid initial \mid terminal \mid legal(a^r) \mid wins(r) \mid does(a^r) \mid$$

$$\neg\varphi \mid \varphi \wedge \psi \mid \bigcirc\varphi \mid \mathsf{K}_r\varphi \mid \mathsf{C}\varphi$$

where $p \in \Phi$, $r \in N$ and $a^r \in A^r$.

Other connectives $\vee$, $\rightarrow$, $\leftrightarrow$, $\top$, $\bot$ are defined by $\neg$ and $\wedge$ in a standard way. The intuitions of all the components inherited from GDL are the same as those in Section 2.1.2.1. The epistemic operators K and C are taken from the Modal Epistemic Logic [Fagin et al., 2003b]. The formula $\mathsf{K}_r\varphi$ is read as "agent r knows φ", and $\mathsf{C}\varphi$ as "φ is common knowledge among all the agents in N".

We use the following abbreviations in the rest of the chapter:

$$\widehat{\mathsf{K}}_r\varphi =_{def} \neg\mathsf{K}_r\neg\varphi \qquad \widehat{\mathsf{C}}\varphi =_{def} \neg\mathsf{C}\neg\varphi \qquad \mathsf{E}\varphi =_{def} \bigwedge_{r\in N} \mathsf{K}_r\varphi$$

where $\widehat{\mathsf{K}}_r$ and $\widehat{\mathsf{C}}$ are the dual operators of K_r and C, respectively. The formula $\widehat{\mathsf{K}}_r\varphi$ says "φ is compatible with agent r's knowledge" and it is similar to $\widehat{\mathsf{C}}\varphi$. The formula $\mathsf{E}\varphi$ says "every agent in N knows φ".

Let us illustrate the intuition of the language with Krieg-mk-games.

Example 4.1 (*continued.*) *The rules of a Krieg-mk-game are specified by EGDL in Figure 4.1 (where $r \in \{\mathsf{x}, \mathsf{o}\}$ and $-r$ represents r's opponent).*

The initial state, each player's winning states, the terminal states and the turn-taking are given by rules 1-4.

1. $initial \leftrightarrow turn(\mathsf{x}) \wedge \neg turn(\mathsf{o}) \wedge \bigwedge_{i,j=1}^{m}(\neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}) \wedge \neg(tried(a^{\mathsf{x}}_{i,j}) \vee tried(a^{\mathsf{o}}_{i,j})))$
2. $wins(r) \leftrightarrow \begin{pmatrix} \bigvee_{i=1}^{m} \bigvee_{j=1}^{m-k+1} \bigwedge_{l=0}^{k-1} p^r_{i,j+l} \vee \bigvee_{i=1}^{m-k+1} \bigvee_{j=1}^{m} \bigwedge_{l=0}^{k-1} p^r_{i+l,j} \\ \vee \bigvee_{i=1}^{m-k+1} \bigvee_{j=1}^{m-k+1} \bigwedge_{l=0}^{k-1} p^r_{i+l,j+l} \vee \bigvee_{i=1}^{m-k+1} \bigvee_{j=k}^{m} \bigwedge_{l=0}^{k-1} p^r_{i+l,j-l} \end{pmatrix}$
3. $teminal \leftrightarrow wins(\mathsf{x}) \vee wins(\mathsf{o}) \vee \bigwedge_{i,j=1}^{m} (p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})$
4. $turn(r) \rightarrow \bigcirc \neg turn(r) \wedge \bigcirc turn(-r)$
5. $legal(noop^r) \leftrightarrow turn(-r)$
6. $legal(a^r_{i,j}) \leftrightarrow turn(r) \wedge \neg p^r_{i,j} \wedge \neg tried(a^r_{i,j})$
7. $\bigcirc p^r_{i,j} \leftrightarrow terminal \vee p^r_{i,j} \vee (does(a^r_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}))$
8. $\bigcirc tried(a^r_{i,j}) \leftrightarrow terminal \vee tried(a^r_{i,j}) \vee (does(a^r_{i,j}) \wedge p^{-r}_{i,j})$
9. $does(a^r_{i,j}) \rightarrow \mathsf{K}_r(does(a^r_{i,j}))$
10. $initial \rightarrow \mathsf{E}\, initial$
11. $(turn(r) \rightarrow \mathsf{E}\, turn(r)) \wedge (\neg turn(r) \rightarrow \mathsf{E} \neg turn(r))$
12. $(p^r_{i,j} \rightarrow \mathsf{K}_r p^r_{i,j}) \wedge (\neg p^r_{i,j} \rightarrow \mathsf{K}_r \neg p^r_{i,j})$
13. $(tried(a^r_{i,j}) \rightarrow \mathsf{K}_r tried(a^r_{i,j})) \wedge (\neg tried(a^r_{i,j}) \rightarrow \mathsf{K}_r \neg tried(a^r_{i,j}))$

FIGURE 4.1: An EGDL description of a Krieg-mk-game.

The preconditions of each action (legality) are specified by rules 5 and 6. The player who has the turn can fill any grid such that (i) it is not filled by herself, and (ii) she has never tried to fill it before. The other player can only do action noop.

Rules 7 and 8 are the combination of the frame axioms and the effect axioms [Reiter, 1991]. Rule 7 states that a grid is marked with a player's symbol in the next state if the player takes the corresponding action at the current state, or the grid has been filled by her symbol before, or the game ends. Similarly, Rule 8 says that an action is tried by a player in the next state if the action is ineffective while still taken by the player at the current state, or this action has been tried before, or the game ends.

The others are the epistemic rules. Rule 9 states each player knows which action she is taking. Rule 10 and Rule 11 say both players know the initial state and the turn-taking, respectively. Rule 12 says that each player knows which grid is filled or not by her own symbol. Similarly, Rule 13 states that each player knows which grid is tried or not by herself.

Let Σ_{KG} *be the set of rules 1-13. It should be noted that rules 11-13 together specify the epistemic relation for each player: two states are indistinguishable for a player if their configurations of the game board are the same in her view.*

4.3.2 The Semantics

Let us now interpret the formulas of EGDL based on the epistemic state transition model.

Definition 4.8. Let M be an ET-model. Given a complete path δ in M, a stage j of δ and a formula $\varphi \in \mathcal{L}_{EGDL}$, we say φ is true (or satisfied) at j of δ under M, denoted by $M, \delta, j \models \varphi$, according to the following definition*:*

$$
\begin{array}{lll}
M, \delta, j \models p & \text{iff} & p \in \pi(\delta[j]) \\
M, \delta, j \models \neg\varphi & \text{iff} & M, \delta, j \not\models \varphi \\
M, \delta, j \models \varphi_1 \wedge \varphi_2 & \text{iff} & M, \delta, j \models \varphi_1 \text{ and } M, \delta, j \models \varphi_2 \\
M, \delta, j \models \mathit{initial} & \text{iff} & \delta[j] = \overline{w} \\
M, \delta, j \models \mathit{terminal} & \text{iff} & \delta[j] \in T \\
M, \delta, j \models \mathit{wins}(r) & \text{iff} & \delta[j] \in g(r) \\
M, \delta, j \models \mathit{legal}(a^r) & \text{iff} & a^r \in L(\delta[j]) \\
M, \delta, j \models \mathit{does}(a^r) & \text{iff} & \theta_r(\delta, j) = a^r \\
M, \delta, j \models \bigcirc\varphi & \text{iff} & \text{if } j < |\delta|, \text{ then } M, \delta, j+1 \models \varphi \\
M, \delta, j \models \mathsf{K}_r\varphi & \text{iff} & \text{for any } \delta' \in \mathcal{P} \text{ with } \delta \approx_r^j \delta',\ M, \delta', j \models \varphi \\
M, \delta, j \models \mathsf{C}\varphi & \text{iff} & \text{for any } \delta' \in \mathcal{P} \text{ with } \delta \approx_N^j \delta'\ M, \delta', j \models \varphi
\end{array}
$$

where $\approx_N^j$ is its transitive closure of $\bigcup_{r \in N} \approx_r^j$.

A formula φ is *globally true* in an ET-model M, written $M \models \varphi$, if $M, \delta, j \models \varphi$ for any $\delta \in \mathcal{P}$ and any $0 \leq j \leq |\delta|$. A formula φ is *valid* in an ET-frame $\mathcal{F}$, written $\mathcal{F} \models \varphi$, if $M \models \varphi$ for any ET-model M based on $\mathcal{F}$. A formula φ is *valid*, written $\models \varphi$, iff $\mathcal{F} \models \varphi$ for any ET-frame $\mathcal{F}$. In particular, a formula φ is *true at a state* w in M, written $M, w \models \varphi$, if it is true for all complete paths going through w, i.e., $M, \delta, j \models \varphi$ for any $\delta \in \mathcal{P}$ and any $j \geq 0$ with $\delta[j] = w$. Finally, let Σ be a set of formulas in $\mathcal{L}_{EGDL}$, then M is a *model* of Σ, if $M \models \varphi$ for all $\varphi \in \Sigma$.

Let us recall that $M_{KG} = (\mathcal{F}_{KG}, \pi_{KG})$ is the ET-model for the Krieg-mk-game, and Σ_{KG} is the set of rules in Figure 4.1. We are now able to show that EGDL provides a sound description for the Krieg-mk-game.

Proposition 4.2. *M_{KG} is a model of Σ_{KG}.*

Proof. Given any complete path δ, any stage t of δ in M_{KG}, we need to verify that each rule is true at t of δ under M_{KG}.

Let us first verify Rule 1. Assume $M_{KG}, \delta, t \models initial$, then $\delta[t] = \overline{w}_{KG}$. And by the definition of π_{KG}, we obtain that $turn(\mathsf{x}) \in \pi_{KG}(\overline{w})$, $turn(\mathsf{o}) \notin \pi_{KG}(\overline{w})$, $p^r_{i,j} \notin \pi_{KG}(\overline{w})$ and $tried(a^r_{i,j}) \notin \pi_{KG}(\overline{w})$ for any $r \in \{\mathsf{x}, \mathsf{o}\}$ and any $1 \leq i, j \leq m$. Thus, $M_{KG}, \delta, t \models turn(\mathsf{x}) \wedge \neg turn(\mathsf{o}) \wedge \bigwedge_{i,j=1}^{m}(\neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}) \wedge \neg(tried(a^{\mathsf{x}}_{i,j}) \vee tried(a^{\mathsf{o}}_{i,j})))$. Conversely, assume $M_{KG}, \delta, t \models turn(\mathsf{x}) \wedge \neg turn(\mathsf{o}) \wedge \bigwedge_{i,j=1}^{m}(\neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}) \wedge \neg(tried(a^{\mathsf{x}}_{i,j}) \vee tried(a^{\mathsf{o}}_{i,j})))$, then by the definition of V_{KG}, we have $\delta[t] = (1, 0, \square, \cdots, \square)$, so $\delta[t] = \overline{w}_{KG}$. Thus, $M_{KG}, \delta, t \models initial$.

Rule 2 and Rule 3 are verified in a similar way of Rule 1.

We now consider Rule 4. Assume $M_{KG}, \delta, t \models turn(r)$, then $t(r) = 1$. If $|\delta| \leq t$, it is straightforward. Otherwise, by the definition of U_{KG} and $\overline{w}_{KG}$, we have $t(-r) = 0$. And again by them we obtain that in $\delta[t+1]$, $t'(r) = 0$ and $t'(-r) = 1$, then $M_{KG}, \delta, t+1 \models \neg turn(r) \wedge turn(-r)$, so $M_{KG}, \delta, t \models \bigcirc(\neg turn(r) \wedge turn(-r))$. Thus, $M_{KG}, \delta, t \models \bigcirc \neg turn(r) \wedge \bigcirc turn(-r)$.

We then consider Rule 6, and Rule 5 is proved in a similar way. Assume $M_{KG}, \delta, t \models legal(a^r_{i,j})$ iff $a^r_{i,j} \in L_{KG}(\delta[t])$ iff $turn(r) \in \pi_{KG}(\delta[t])$, $p^r_{i,j} \notin \pi_{KG}(\delta[t])$ and $tried(a^r_{i,j}) \notin \pi_{KG}(\delta[t])$ (by definitions of L_{KG} and π_{KG}) iff $M_{KG}, \delta, t \models turn(r) \wedge \neg p^r_{i,j} \wedge \neg tried(a^r_{i,j})$.

We now verify Rule 7, and Rule 8 is verified in a similar way. Assume $M_{KG}, \delta, t \models terminal \vee p^r_{i,j} \vee (does(a^r_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}))$. We next prove by three cases.
(1) If $M_{KG}, \delta, t \models terminal$, then $t = |\delta|$, so it is trivial that $M_{KG}, \delta, t \models \bigcirc p^r_{i,j}$;
(2) If $M_{KG}, \delta, t \models p^r_{i,j}$, by the definition of U_{KG}, we have $M_{KG}, \delta, t+1 \models p^r_{i,j}$, so $M_{KG}, \delta, t \models \bigcirc p^r_{i,j}$;
(3) If $M_{KG}, \delta, t \models does(a^r_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})$, then by the definition of U_{KG}, we have $M_{KG}, \delta, t+1 \models p^r_{i,j}$, so $M_{KG}, \delta, t \models \bigcirc p^r_{i,j}$.
Thus, in all cases we have $M_{KG}, \delta, t \models \bigcirc p^r_{i,j}$.
Conversely, assume $M_{KG}, \delta, t \not\models terminal \vee p^r_{i,j} \vee (does(a^r_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}))$, then $|\delta| > t$, $M_{KG}, \delta, t \not\models p^r_{i,j}$ and $M_{KG}, \delta, t \not\models (does(a^r_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}))$. And by the definition of U_{KG}, we obtain $M_{KG}, \delta, t \not\models \bigcirc p^r_{i,j}$.

We finally verify the epistemic rules 9-13. Rule 9 directly follows from the assumption of M_{KG}, and Rule 10 follows from the fact that $R^{KG}_r(\overline{w}_{KG}) = \{\overline{w}_{KG}\}$ for any $r \in \{\mathsf{x}, \mathsf{o}\}$. It is straightforward for rules 11-13 by the definition of R^{KG}_r. $\square$

It follows that these game rules are common knowledge among two players, which is just what we expect.

Corollary 4.3. *$M_{KG} \models \mathsf{C}\varphi$ for all $\varphi \in \Sigma_{KG}$.*

4.4 Epistemic and Strategic Reasoning

In this section, we demonstrate the expressive power and flexibility of EGDL by showing how it allows us to specify epistemic properties and reason about agents' knowledge during game playing.

4.4.1 Epistemic Properties

The introduction of imperfect information raises new epistemic properties of a game. For instance, to make a game playable, each player should always know her own legal actions in the course of the game. This property as well as other well-known properties can be naturally formulated by EGDL. Given $r \in N$ and $a^r \in A^r$,

(1) $initial \to \mathsf{C}\, initial$ (2) $legal(a^r) \to \mathsf{K}_r(legal(a^r))$

(3) $does(a^r) \to \mathsf{K}_r(does(a^r))$ (4) $wins(r) \to \mathsf{K}_r(wins(r))$

(5) $terminal \to \mathsf{C}\, terminal$

Formulas (1) and (5) express that the initial state, the terminal states are common knowledge among agents, respectively. Formula (2) says that each agent knows her own legal actions. In ATEL, this is a required semantic property yet with no syntactic expression [Ågotnes, 2006]. Formula (3) asserts that each agent is aware of her own actions. This is called the "uniform" property of actions (strategies) also with no syntactic counterpart in ATEL [Jamroga and van der Hoek, 2004, van der Hoek and Wooldridge, 2003]. Finally, Formula (4) specifies that each agent should know her winning result.

The above epistemic properties are precisely characterised by indistinguishable complete paths as follows:

Proposition 4.4. *Let $\mathcal{F}$ be an ET-frame. Then*

1. $\mathcal{F} \models initial \to \mathsf{C}\, initial$ *iff for all* $\delta, \delta' \in \mathcal{P}$ *and any* $j \in \mathbb{N}$, *if* $\delta \approx^j_N \delta'$, *then* ($\delta[j] = \overline{w}$ *iff* $\delta'[j] = \overline{w}$).

2. $\mathcal{F} \models legal(a^r) \to \mathsf{K}_r(legal(a^r))$ *iff for all* $\delta, \delta' \in \mathcal{P}$ *and any* $j \in \mathbb{N}$, *if* $\delta \approx^j_r \delta'$, *then* ($a^r \in L(\delta[j])$ *iff* $a^r \in L(\delta'[j])$).

3. $\mathcal{F} \models does(a^r) \rightarrow \mathsf{K}_r(does(a^r))$ *iff for all* $\delta, \delta' \in \mathcal{P}$ *and any* $j \in \mathbb{N}$, *if* $\delta \approx_r^j \delta'$, *then* $(\theta_r(\delta, j) = a^r$ *iff* $\theta_r(\delta', j) = a^r)$.

4. $\mathcal{F} \models wins(r) \rightarrow \mathsf{K}_r(wins(r))$ *iff for all* $\delta, \delta' \in \mathcal{P}$ *and any* $j \in \mathbb{N}$, *if* $\delta \approx_r^j \delta'$, *then* $(\delta[j] \in g(r)$ *iff* $\delta'[j] \in g(r))$.

5. $\mathcal{F} \models terminal \rightarrow \mathsf{C}\, terminal$ *iff for all* $\delta, \delta' \in \mathcal{P}$ *and any* $j \in \mathbb{N}$, *if* $\delta \approx_N^j \delta'$, *then* $(\delta[j] \in T$ *iff* $\delta'[j] \in T)$.

Proof. 1. Assume there are some model M based on $\mathcal{F}$, two complete paths $\delta, \delta' \in \mathcal{P}$ and some $j \in \mathbb{N}$ such that $\delta \approx_N^j \delta'$, $\delta[j] = \overline{w}$ and $\delta'[j] \neq \overline{w}$, then $M, \delta, j \models initial$ and $M, \delta', j \not\models initial$, so $M, \delta, j \not\models \mathsf{C}\, initial$. Then $M, \delta, j \not\models initial \rightarrow \mathsf{C}\, initial$, so $\mathcal{F} \not\models initial \rightarrow \mathsf{C}\, initial$. Conversely, assume $\mathcal{F} \not\models initial \rightarrow \mathsf{C}\, initial$, then there are some model M based on $\mathcal{F}$, some $\delta \in \mathcal{P}$ and some $j \in \mathbb{N}$ such that $M, \delta, j \not\models initial \rightarrow \mathsf{C}\, initial$, then $M, \delta, j \models initial$ and $M, \delta, j \not\models \mathsf{C}\, initial$, so $\delta[j] = \overline{w}$, and there is some $\delta \in \mathcal{P}$ such that $\delta \approx_N^j \delta'$ and $M, \delta', j \not\models initial$. Thus, $\delta'[j] \neq \overline{w}$.

The other statements are proved in a similar way. $\square$

Obviously, not all games with imperfect information satisfy these epistemic properties. For instance, epistemic property (5) does not hold for Krieg-mk-games. Consider the two reachable states depicted in Figure 4.2. They are indistinguishable for player o in Krieg-Tictactoe. Yet the left one is a terminal state while the right one is not. Indeed, according to the game rules, a Krieg-mk-game satisfies all the other properties.

o	x	o
	x	
	x	

o		o
	x	
x		x

Figure 4.2: The indistinguishable states for o.

Observation 4.5. Formulas (1)-(4) are globally true in M_{KG}.

Proof. Formula (1) follows from Rule 10. Formula (2) follows from rules 5-6 and 11-13. Formula (3) follows from Rule 9, and formula (4) follows from rules 2 and 11-13. □

Furthermore, we have the following result in terms of the underlying structure of a Krieg-mk-game.

Observation 4.6. Formulas (1)-(4) are valid in $\mathcal{F}_{KG}$.

It should be noted that each EGDL-formula may be interpreted as a property of a game. Typically, globally true formulas describe properties of a particular game, such as the rules for a Krieg-mk-game, while valid formulas specify general properties of a class of games, and thus can be used to classify games. For instance, different from Krieg-style board games, most card games have the epistemic property (5).

4.4.2 Strategic Reasoning

Let us now show how to use EGDL to reason about agents' knowledge and actions based on game rules. In the context of imperfect information, epistemic reasoning is closely related to strategic reasoning. To start with, the following proposition shows that EGDL is suitable for reasoning about players' knowledge, as it is a conservative extension of the standard Epistemic Modal Logic $S5^C_n$ [Fagin et al., 2003b].

Proposition 4.7. *Given an EGDL-formula φ without involving the operator $\bigcirc$ and the pre-defined propositions, φ is valid in EGDL iff it is valid in $S5^C_n$.*

Proof. We prove this result in two steps. Let us first associate a multi-agent epistemic model E with an ET-model M_E as follows:

Given a multi-agent epistemic model $E = (S, \{\mathcal{E}_i\}_{i\in Agt}, V)$ with a set Agt of agents and a set Atm of propositional variables, we define the game signature

$\mathcal{S} = (Agt, \mathcal{A}, Atm)$ with the same set Agt of agents and the same set Atm of propositional variables such that $A^x = \{select(s) \mid s \in S\}$ and $A^r = \{noop^r\}$ for any $r \in Agt \backslash \{x\}$.

Based on this, we construct an associate ET-model $M_E = (W, \overline{w}, T, \{R_r\}_{r \in N}, L, U, g, \pi)$ as follows:

- $W = S \cup \{\overline{w}\}$ where $\overline{w} \notin S$ is the initial state.
- $T = g = S$.
- $s_1 R_r s_2$ iff $s_1 \mathcal{E}_r s_2$ for any $s_1, s_2 \in S$ and $r \in Agt$.
- $L = W \backslash T \times \mathcal{A}$.
- For any $d \in D$ and any $w \in W$,
  - if $w \in S$, $U(w, d) = w$;
  - if $w = \overline{w}$, $U(\overline{w}, d) = s$ iff $d(x) = select(s)$.
- For any $w \in W$,
  - if $w \in S$, $p \in \pi(w)$ iff $p \in V(w)$ for every $p \in Atm$;
  - if $w = \overline{w}$, $\pi(w) \cap Atm = \emptyset$.

We write d^s for joint action d with player x selects state s, i.e., $d(x) = select(s)$. Then the successor state $s = U(\overline{w}, d^s)$ corresponds to a complete path $\overline{w} \xrightarrow{d^s} s$, denoted by δ^s. Let $\mathcal{P}$ be the set of all complete paths so obtained, i.e., $\mathcal{P} = \{\delta^s \mid s \in S\}$. By a routine induction on the structure of φ, we obtain the following claim.

Claim 4.8. Given a multi-agent epistemic model E and a state s in E,

$$E, s \models_{\mathrm{S5}_n^C} \varphi \text{ iff } M_E, \delta^s, 1 \models_{\mathrm{EGDL}} \varphi.$$

Next we transform an ET-model M into a multi-agent epistemic model E_M in the following way.

Fix an ET-model $M = (W, \overline{w}, T, \{R_r\}_{r\in N}, L, U, g, \pi)$ with a game signature $\mathcal{S} = (N, \mathcal{A}, \Phi)$. We construct $E_M = (S, \{\mathcal{E}_r\}_{r\in N}, V)$ with the same set of agents in M and the same set of propositional variables in Φ. Let $\delta[0, j]$ denote the *initial segment* of complete path δ up to stage j. For convenience, we call it a *history.* Then the components of E_M are defined as follows:

- $S = \{\delta[0, j] \mid \delta \in \mathcal{P}, j \geq 0\}$ is the set of all histories in M.
- for all $\delta[0, j], \delta'[0, j'] \in S$ and $r \in N$, $\delta[0, j]\mathcal{E}_r\delta'[0, j']$ iff $j = j'$ and $\delta \approx^j_r \delta'$.
- $p \in V(\delta[0, j])$ iff $p \in \pi(\delta[j])$ for any $p \in \Phi$.

By induction on the structure of formula φ, we have the following claim.

Claim 4.9. Given an ET-model M, a complete path δ and a stage $j \geq 0$,

$$M, \delta, j \models_{\mathrm{EGDL}} \varphi \text{ iff } E_M, \delta[0, j] \models_{\mathrm{S5}^C_n} \varphi.$$

Thus, the result follows directly from the two claims. □

This result indicates that EGDL is sufficient to provide a static characterization of agents' knowledge at a certain stage. For instance, with $S5^C_n$, we can derive the following formulas from the rules of a Krieg-*mk*-game.

Observation 4.10.

1. $M_{KG} \models turn(r) \to \mathsf{C}\, turn(r)$
2. $M_{KG} \models \mathsf{K}_r(\mathsf{K}_{-r}p^{-r}_{i,j} \vee \mathsf{K}_{-r}\neg p^{-r}_{i,j})$
3. $M_{KG} \models \mathsf{K}_r tried(a^r_{i,j}) \to \mathsf{K}_r p^{-r}_{i,j}$

Proof. Clause 1 follows from Rule 11 and Induction Rule (RC1). Clause 2 follows from Rule 12 and Theorem 3.1.1(d). The last clause follows from rules 7-8 and A2, Theorem 3.1.1 in [Fagin et al., 2003b]. □

Clause 1 says the turn-taking is common knowledge. Clause 2 says a player knows the opponent knows whether or not a grid is filled by herself. The last one says if a player knows she has tried an action, then she knows the corresponding grid has been filled by the opponent. The last two properties are important when players gather information.

Furthermore, with the full expressive power of the language, we can use EGDL to specify agents' knowledge of particular game features and reason about how agent's knowledge changes as a game progresses.

Observation 4.11.

1. $M_{KG} \models \mathsf{K}_r p^r_{i,j} \rightarrow \mathsf{K}_r \bigcirc p^r_{i,j}$
2. $M_{KG} \models \mathsf{K}_r \bigcirc tried(a^r_{i,j}) \rightarrow \bigcirc \mathsf{K}_r tried(a^r_{i,j})$
3. $M_{KG} \models initial \rightarrow \mathsf{C}\,(\bigwedge^m_{i,j=1} legal(a^{\times}_{i,j}) \wedge legal(noop^{\circ}))$
4. $M_{KG} \models does(a^r_{i,j}) \rightarrow \bigcirc \mathsf{K}_r(p^r_{i,j} \vee tried(a^r_{i,j}))$

Proof. Clause 1 follows from Rule 7. Clause 2 follows from rules 8 and 13. Clause 3 follows from rules 1, 5-6 and 10. The last clause follows from rules 7-8. □

Intuitively, clause 1 says that if a player knows a grid has been filled by herself, then she still knows this fact at the next state. Clause 2 says that a player is able to remember the grid she has tried to fill before. Clause 3 says that at the initial state the legal actions are common knowledge among two players, and Clause 4 expresses that if a player takes an action now, then at the next state she will know either the corresponding grid has been filled by her symbol, or she has tried that action.

Most importantly, the interactions of actions and knowledge can be naturally formulated using EGDL. Specifically, they interact in three different ways:

(i) Knowledge is necessary for an agent to perform an action, which may be formulated as $does(a^r) \rightarrow \mathsf{K}_r\varphi$. For instance, in a Krieg-mk-game, with partial observation, a player might take an ineffective action by trying to fill a grid which has been filled by the opponent. Then we say a player r takes a good action $a^r_{i,j}$, written $good(a^r_{i,j})$, if it is effective. It follows that, to take a good action, a player needs to know the grid she attempts to fill is empty. Formally, $M_{KG} \models good(a^r_{i,j}) \rightarrow \mathsf{K}_r(\neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}))$.

(ii) Performing an action may increase an agent's knowledge, which may be specified by $does(a^r) \rightarrow \bigcirc\mathsf{K}_r\varphi$. For example, if a player takes an ineffective action, then she would know the corresponding grid has been filled by the opponent. Consider the following complete path

$$\delta = \overline{w} \xrightarrow{\langle a^{\mathsf{x}}_{2,2}, noop^{\mathsf{o}}\rangle} w_1 \xrightarrow{\langle noop^{\mathsf{x}}, a^{\mathsf{o}}_{2,2}\rangle} w_2 \xrightarrow{\langle a^{\mathsf{x}}_{1,1}, noop^{\mathsf{o}}\rangle} w_3 \cdots$$

At stage 2 after player o tries to fill grid (2,2), by Rule 7 and Rule 13, she knows that the grid has been filled by player x. Thus, $M_{KG}, \delta, 1 \models does(a^{\mathsf{o}}_{2,2}) \rightarrow \bigcirc\mathsf{K}_{\mathsf{o}}(tried(a^{\mathsf{o}}_{2,2}) \wedge p^{\mathsf{x}}_{2,2})$.

(iii) An agent makes her choice of actions based on her knowledge, which may be captured by $\mathsf{K}_r\varphi \rightarrow does(a^r)$. Let us consider the following two basic actions:

$$attack^r =_{def} \mathsf{K}_r(does(a^r_{i,j}) \wedge \bigcirc wins(r)) \rightarrow does(a^r_{i,j})$$
$$block^r =_{def} \mathsf{K}_r\bigcirc(does(a^{-r}_{i,j}) \wedge \bigcirc wins(-r)) \rightarrow does(a^r_{i,j})$$

Intuitively, *attack* says if a player knows that filling a grid leads to win, then she should fill that grid. Instead, *block* says if a player knows her opponent makes to win by filling a grid in the next state, then the player must fill that grid at the current state to avoid an immediate loss.

4.5 Model Checking

To systematically check whether an imperfect information game satisfies a given property, we now investigate the model-checking problem for EGDL and develop a model-checking algorithm for this logic.

The *model checking problem* for EGDL, denoted by EGDL-MC, is the following: Given an EGDL-formula φ, an ET-model M, a path δ of M and a stage j on δ, determine whether $M, \delta, j \models \varphi$ or not. In principle, two variants of EGDL-MC can be defined as follows: Given an ET-model M, a state w of M and an EGDL-formula φ, determine whether $M, w \models \varphi$ and determine whether $M \models \varphi$. It should be noted that all the bounds presented in this section remain true for these variants. Proofs are similar to those of EGDL-MC, or can be obtained by simple reductions to/from EGDL-MC.

Before presenting the results, let us first recall some notations in the theory of computational complexity. By PTIME (respectively, NP) we denote the class of languages (i.e., decision problems) decidable in polynomial-time deterministic (respectively, nondeterministic) Turing machines. Let Δ_2^p (respectively, Θ_2^p) denote the class of languages each of which is decidable in a polynomial-time deterministic Turing machine with a polynomial (respectively, logarithmic) number of queries to an NP language as an oracle (or simply, an NP-oracle). It is worth noting that both Δ_2^p and Θ_2^p lie in the 2^{nd} level of the polynomial hierarchy; both of them contain NP and coNP, i.e. the complement of NP. It remains open whether Θ_2^p is a proper subclass of Δ_2^p or not.

Let us first consider the upper bound of the complexity for model-checking. Our goal is to show the following bound.

Theorem 4.12. *EGDL-MC is in Δ_2^p.*

To prove this upper bound, according to the definition of Δ_2^p, we need to show that there is a polynomial-time deterministic Turing machine $\mathcal{M}$ with an NP-oracle such that $\mathcal{M}$ solves the model-checking problem for EGDL. To this end, let us start with a simple property of EGDL.

Let φ be an EGDL-formula, and $M = (\mathcal{F}, \pi)$ be an ET-model over $\mathcal{S}$. Take ψ to be any subformula of φ of the form $\oplus\vartheta$, where $\oplus$ is either C or K_r for some $r \in N$. We introduce a fresh propositional variable p_ψ for ψ. Let M_ψ be the ET-model $(\mathcal{F}, \pi_\psi)$ where π_ψ is a valuation function defined as follows: For each state w of M,

$$\pi_\psi(w) = \begin{cases} \pi(w) \cup \{p_\psi\} & \text{if } M, w \models \psi; \\ \pi(w) & \text{otherwise.} \end{cases}$$

Let φ_ψ denote the formula obtained from φ by replacing ψ by p_ψ. Then, by the definition of semantics for EGDL, the following property is true.

Lemma 4.13. *For every path δ of M and every stage j on δ, it holds that $M, \delta, j \models \varphi$ iff $M_\psi, \delta, j \models \varphi_\psi$.*

Thus, by applying the above lemma, the epistemic operators can be eliminated from the formula in a recursive way. For any EGDL-formula without involving any epistemic operator, we can show that its model-checking problem is tractable.

Lemma 4.14. *The following problem is in PTIME: Given an ET-model M, a path δ of M, a stage j on δ and an EGDL-formula φ without involving any epistemic operators, determine whether $M, \delta, j \models \varphi$ or not.*

Proof. It is routine to design an algorithm to traverse over the parse tree of φ and work in the following way: For each visit of a node ψ (which is a subformula of φ) in the tree, the algorithm first evaluates the truth values of all the proper subformulas of ψ; with these truth values, the truth value of ψ can be then easily obtained by the definition of semantics. As the algorithm visits each node at most once, and the number of nodes in the tree is not greater than the size of φ, such an algorithm can be clearly implemented in a polynomial-time deterministic Turing machine, which is as desired. □

To construct the model M_ψ, the truth value of ψ at a given state under M is needed to be evaluated. To simplify the question, let us first consider a simple case as follows:

Lemma 4.15. *The following problem is in NP: Given an ET-model M, a state w of M and an EGDL-formula $\oplus\varphi$ where $\oplus \in \{\widehat{\mathsf{C}}\} \cup \{\widehat{\mathsf{K}}_r : r \in N\}$ and φ does not involve any epistemic operators, determine whether $M, w \models \oplus\varphi$ or not.*

Proof. Let k be the number of occurrences of $\bigcirc$ in φ. Take δ be any path of M, and $i \geq 0$ be a stage . According to the definition, it is routine to show that $M, \delta, i \models \varphi$ if, and only if, $M, \delta[0, k], i \models \varphi$. Thus, one can easily verify that the nondeterminitic algorithm *istrueE* (see Algorithm 4.5.1) is sound for the problem given in the lemma. By Lemma 4.14, we can see that *istrueE* can be implemented in a polynomial time nondeterminstic Turing machine. Note that k is not greater than the size of φ, which means that the machine only needs to guess $O(n \log n)$ bits of information, where n is the size of φ. This then proves the lemma. □

To eliminate all the epistemic operators, it remains to consider the formulas with nested epistemic operator. With this complexity result for the non-nested case, we are now able to design an algorithm for the general case. Roughly speaking, the idea is to carry out the elimination of epistemic operator in a bottom-up way. As we can see in Algorithm 4.5.2, such an idea is implemented in the algorithm *elimeop*.

It's not hard to check that the algorithm can be implemented in a polynomial-time deterministic Turing machine, but with an NP-oracle. Here the procedure of checking $N_0, w \models \widehat{\oplus}\neg\psi_0$ is used as the NP-oracle. By Lemma 4.15, the checking is in NP. In addition, the algorithm *elimeop* visits each subformula of φ at most

Algorithm 4.5.1: $istrueE(M, w, \varphi)$

Input : an ET-model M, a state w of M, and an EGDL-formula of the form $\oplus\varphi$ where $\oplus \in \{\widehat{\mathsf{C}}\} \cup \{\widehat{\mathsf{K}}_r : r \in N\}$ and no epistemic operator occurs in φ

Output: **true** if $M, w \models \oplus\varphi$, and **false** otherwise

begin
  if $\oplus$ *is* $\widehat{\mathsf{K}}_r$ *for some* $r \in N$ **then**
    $R \leftarrow R_r$;
  else
    $R \leftarrow$ the transitive closure of $\bigcup_{r \in N} R_r$;
  end
  $k \leftarrow$ the number of occurrences of $\bigcirc$ in φ;
  guess a path δ of M of a length $\leq k$;
  if δ *is complete* **and** $(w, \delta[0]) \in R$ **then**
    return $M, \delta, 1 \models \varphi$;
  else
    return false;
  end
end

once, and the number of subformulas of φ is not greater than the size of φ. These assure that the Turing machine will terminate in a polynomial number of stages.

With this algorithm, we then devise an algorithm *mc* for the model-checking problem of EGDL such that, given any proper M, δ, j and φ as input, *mc* works as follows:

- First, *mc* calls the algorithm $elimeop(M, \varphi)$, and let (M_0, φ_0) be the results of this call.
- Next, *mc* checks whether $M_0, \delta, j \models \varphi_0$ or not, and return "true" if it holds, "false" otherwise.

By Lemma 4.13 and the definition of algorithm *elimeop*, it is not difficult to verify that $M, \delta, j \models \varphi$ if, and only if, $M_0, \delta, j \models \varphi_0$. This assures the soundness of the algorithm *mc*. On the other hand, by the previous analysis, the first stage can be implemented in a polynomial-time determinitic Turing machine with an NP-oracle; by Lemma 4.14, the second stage can be done in PTIME. Thus, the algorithm *mc*

Algorithm 4.5.2: *elimeop*(M, φ)

Input : an ET-model M and an EGDL-formula φ
Output: an ordered pair (M_0, φ_0)

begin
  switch φ **do**
    case φ *is atomic*
      $M_0 \leftarrow M$;
      $\varphi_0 \leftarrow \varphi$;

    case φ *is of the form* $\oplus\psi$*, where* $\oplus \in \{\neg, \bigcirc\}$
      $(N_0, \psi_0) \leftarrow elimeop(M, \psi)$;
      $M_0 \leftarrow N_0$;
      $\varphi_0 \leftarrow \oplus\psi_0$;

    case φ *is of the form* $\psi \wedge \chi$
      $(N_0, \psi_0) \leftarrow elimeop(M, \psi)$;
      $(N_0, \chi_0) \leftarrow elimeop(N_0, \chi)$;
      $M_0 \leftarrow N_0$;
      $\varphi_0 \leftarrow \psi_0 \wedge \chi_0$;

    case φ *is of the form* $\oplus\psi$*, where* $\oplus \in \{\mathsf{C}\} \cup \{\mathsf{K}_r : r \in N\}$
      $(N_0, \psi_0) \leftarrow elimeop(M, \psi)$;
      $\pi \leftarrow$ the valuation function of N_0;
      for all w **in** W **do**
        if $N_0, w \models \widehat{\oplus}\neg\psi_0$ *is false, where* $\widehat{\oplus} \in \{\widehat{\mathsf{C}}, \widehat{\mathsf{K}}_r\}$ **then**
          $\pi(w) \leftarrow \pi(w) \cup \{p_{\oplus\psi}\}$;
        end
      end
      $M_0 \leftarrow$ the model obtained from N_0 by replacing the valuation function with π;
      $\varphi_0 \leftarrow p_{\oplus\psi}$;

  endsw
  return (M_0, φ_0);
end

can be implemented in a polynomial-time determinitic Turing machine with an NP-oracle, which proves Theorem 4.12.

Next we will identify a lower bound of the complexity of model-checking for EGDL. This bound shows that the above algorithm is nearly optimal. Let us present the bound.

Theorem 4.16. *EGDL-MC is Θ_2^p-hard.*

To prove this, we reduce the validity problem for Carnap's modal logic $\mathbb{C}$, which has been proved to be Θ_2^p-complete by [Gottlob, 1995], to the problem stated in the proposition. Below let us first recall some notions for logic $\mathbb{C}$.

The logic $\mathbb{C}$ is armed with a standard language of the basic modal logic. The validity in $\mathbb{C}$ is defined as follows:

- For each propositional formula φ, $\models_{\mathbb{C}} \varphi$ iff φ is valid in the classical propositional logic $\mathbb{P}$.
- For each formula φ of form $\Box\psi$, $\models_{\mathbb{C}} \varphi$ iff ψ^+ is valid in $\mathbb{P}$.
- For each formula φ of other forms, $\models_{\mathbb{C}} \varphi$ iff φ^+ is valid in $\mathbb{P}$.

In above, by φ^+ we denote the formula obtained from φ by replacing each occurrence of subformulas ψ of the form $\Box\vartheta$ by $\top$ if $\models_{\mathbb{C}} \psi$, and $\bot$ otherwise.

It should be noted that in $\mathbb{C}$, as usual, $\Diamond$ is defined as $\neg\Box\neg$. Now we are in the position to present the reduction. Let φ be a modal formula. We want to construct an ET-model M, a path δ, a stage j on δ and an EGDL-formula ψ such that $\models_{\mathbb{C}} \varphi$ iff $M, \delta, j \models \psi$. The feasibility of such a reduction is based on the following observations: Carnap's modal operators $\Box$ can be simulated by K, and each valuation in $\mathbb{P}$ can be encoded by a complete path.

Next let us define the construction. Suppose $p_1, \ldots, p_n$ is an enumeration of all the propositional variables that occur in φ. Set $\mathcal{S} = (N, \mathcal{A}, \Phi)$ to be a game signature such that $|N| = 1$, $\mathcal{A} = \{+, -\}$ and $\Phi = \{q\}$ is a singleton, where q is a fresh propositional variable. Let $\mathcal{F}$ be an ET-frame $(W, \overline{w}, T, R, L, U, g)$ over $\mathcal{S}$ such that

- $W = \{\overline{w}\} \cup \{w_i^+, w_i^- : 1 \leq i \leq n\}$;

- $T = \{w_n^+, w_n^-\}$;
- $R = W \times W$;
- $L = W \backslash T \times \mathcal{A}$;
- $U(\overline{w}, *) = w_1^*$ for $* \in \{+, -\}$, and
  $U(w_i^*, \star) = w_{i+1}^\star$ for $*, \star \in \{+, -\}$ and $1 \leq i < n$.

With these settings, let us now show how valuations in the classical propositional logic can be related to complete paths of M. Let $v : \{p_i : 1 \leq i \leq n\} \rightarrow \{0, 1\}$ be a valuation. We construct a complete path, denoted by λ_v, of M as follows:

$$\overline{w} \xrightarrow{s_1} w_1^{s_1} \xrightarrow{s_2} w_2^{s_2} \xrightarrow{s_3} \cdots \xrightarrow{s_n} w_n^{s_n} \tag{4.1}$$

where, for each i with $1 \leq i \leq n$, s_i denotes "+" if $v(p_i) = 1$, and "−" otherwise. On the other hand, take λ to be any complete path of M. It is clear that λ must be of the form (4.1). Now we define a valuation, denoted v_λ, such that $v_\lambda(p_i)$ is 1 if s_i is "+", and 0 otherwise.

Let $\pi : W \rightarrow 2^\Phi$ be a valuation function based on $\mathcal{F}$ such that

$$\pi(w) = \begin{cases} \emptyset & \text{if } w \text{ is } \overline{w} \text{ or } w_i^- \text{ where } 1 \leq i \leq n; \\ \{q\} & \text{if } w \text{ is } w_i^+ \text{ where } 1 \leq i \leq n. \end{cases}$$

Let M be the ET-model $(\mathcal{F}, \pi)$. Clearly, given any valuation v in $\mathbb{P}$, we have $v(p_i) = 1$ if, and only if, $M, \lambda_v, 0 \models \bigcirc^i q$. Based on this property, we then define a transformation Tr from the formulas in $\mathbb{C}$ to EGDL-formulas as follows:

$$\mathrm{Tr}(\varphi) = \begin{cases} \bigcirc^i q & \text{if } \varphi = p_i \text{ where } 1 \leq i \leq n; \\ \neg\mathrm{Tr}(\psi) & \text{if } \varphi = \neg\psi; \\ \mathrm{Tr}(\psi) \wedge \mathrm{Tr}(\chi) & \text{if } \varphi = \psi \wedge \chi; \\ \mathsf{K}(\mathit{initial} \rightarrow \mathrm{Tr}(\psi)) & \text{if } \varphi = \Box\psi. \end{cases}$$

It is obvious that, with a natural approach for encoding, both M and $\mathrm{Tr}(\varphi)$ are of polynomial sizes w.r.t. φ. Thus, to prove Theorem 4.16, it suffices to show the following lemma.

Lemma 4.17. *Let φ be a modal formula, and let δ be any path of M. Then $\models_{\mathbb{C}} \varphi$ iff $M, \delta, 0 \models \mathsf{K}(initial \rightarrow \mathrm{Tr}(\varphi))$.*

Proof. Let us first consider the "if" direction of the claim. Assume that $\models_{\mathbb{C}} \varphi$ is not true. By definition, φ must be invalid in $\mathbb{P}$. This means that there exists a valuation v in $\mathbb{P}$ such that $v(\varphi) = 0$. By a routine induction on the structure of φ, we obtain that $M, \lambda, 0 \models \neg\mathrm{Tr}(\varphi)$. From this we have $M, \delta, 0 \models \neg\mathsf{K}(initial \rightarrow \mathrm{Tr}(\varphi))$, as desired.

For the converse, let us assume $M, \delta, 0 \models \neg\mathsf{K}(initial \rightarrow \mathrm{Tr}(\varphi))$. There is thus a complete path λ of M such that $M, \lambda, 0 \models \neg\mathrm{Tr}(\varphi)$. Again by an induction on φ, we have $v_\lambda(\varphi) = 0$, which means that φ is invalid. By definition, $\models_{\mathbb{C}} \varphi$ must be false. This completes the proof. $\square$

4.6 Summary

In this chapter, we have presented a logical framework for representing and reasoning about imperfect information games with imperfect recall players. We have demonstrated that the framework allows us to represent game rules, formalize epistemic properties, specify the interactions of knowledge and actions as well as reason about agents' knowledge during game playing. We have also investigated the model-checking problem for the logic. These results show that the framework has made a reasonable compromise between expressive power and computational efficiency.

Most of related work has been discussed in Section 1.2. Besides that, the following is also worth mentioning. Ruan and Thielscher [2011] study the epistemic structure and expressiveness of GDL-II in terms of epistemic modal logic. Yet they only provide a static characterization of players' knowledge at a certain stage without involving the temporal dimension. Haufe and Thielscher [2012] develop an automated reasoning method to deal with epistemic properties for GDL-II. Different from ours, their method is restricted to positive-epistemic formulas. Our underlying language is from [Zhang and Thielscher, 2015b]. It is originally proposed for reasoning about strategies of asynchronous games with perfect information, while we study its epistemic extension for representing and reasoning about synchronous games with imperfect information. Finally, it should be noted that EGDL has similarities with epistemic temporal logics such as CKL_m [Halpern et al., 2004], but they are significantly different in the following ways: (i)With $does(.)$ operator, EGDL can express actions and their effects, thus it is essentially a logic for reasoning about actions, while epistemic temporal logics are not. (ii) EGDL contains a single temporal operator ("next"), and can only represent finite steps of time. (iii) Model checking for EGDL is in Δ_2^p, while, for epistemic temporal logics, it is PSPACE-hard as the complexty for the underlying logic linear temporal time logic is already PSPACE-complete [Sistla and Clarke, 1985].

Directions of future research are manifold. As we have mentioned, besides imperfect recall, the framework is flexible enough to specify other memory types. We plan to study properties of these memory types, and further investigate the interplay between agents' memory abilities and their knowledge; We also want to investigate the satisfiability problem and the axiomatization of EGDL based on the current literature [Halpern et al., 2004, Zhang and Thielscher, 2015a]; In addition, it would be interesting to explore the dynamic epistemic extension of EGDL so as to study the update of players' knowledge during game play [van Ditmarsch et al., 2007].

Chapter 5

Knowledge Sharing in Coalitions

With imperfect information, agents' abilities are associated with their knowledge. For instance, assuming a few agents are trying to open a safe, only the ones who know the code have the ability to open the safe. One question arises naturally: *which kind of group knowledge is required for a group to achieve some goal in the context of imperfect information?* [Herzig, 2015]. Most of the time when a set of agents forms a coalition, their cooperation is not merely limited to acting together, but, more importantly, sharing their knowledge when acting. Safe opening is an example. Based on this consideration, we assume that whenever a set of agents forms a coalition to achieve a goal, they share their knowledge before acting. With this assumption, this chapter proposes a variant of semantics for ATL with imperfect information to investigate the interplay between knowledge shared by a group of agents and its coalitional abilities. We also show that this semantics is sufficient to preserve the plausible properties of coalitional abilities in Section 3.1.

5.1 A Motivating Example

Let us begin with the following example to highlight our motivation to study coalitional abilities under the assumption of knowledge sharing within coalitions.

Example 5.1. *Figure 5.1 depicts a variant of the shell game [Bulling et al., 2014] with three players: the shuffler s, the guessers g_1 and g_2. Initially the shuffler places a ball in one of the two shells (the left (L) or the right (R)). Guesser g_1 can observe which action the shuffler does, while guesser g_2 cannot. A guesser or a coalition of two wins, if she picks up the shell containing the ball. We assume that only guesser g_2 can choose the shell (the left (l) or the right (r)) and guesser g_1 can only take action noop (n).*

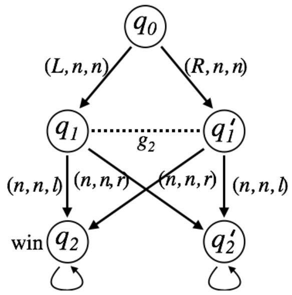

FIGURE 5.1: Model $\mathcal{T}_1$.

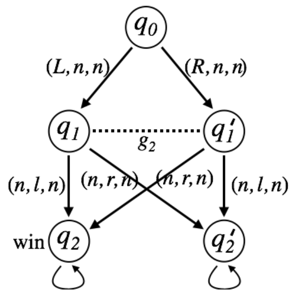

FIGURE 5.2: Model $\mathcal{T}_2$.

The tuple $(\alpha_1, \alpha_2, \alpha_3)$ represents an action profile,.i.e, action α_1 of shuffler s, action α_2 of guesser g_1, and action α_3 of guesser g_2. The dotted line represents g_2's indistinguishability relation: reflexive loops are omitted. State q_2 is labelled with the proposition win.

Clearly, g_1 knows the location of the ball but cannot choose. Instead g_2 does not know where the ball is, though he can choose the shell. Thus, neither g_1 nor g_2 can win this game individually. But if g_1 and g_2 form a coalition, it should follow that by sharing their knowledge they can cooperate to win. However, according to most of existing semantics for ATL with imperfect information [Jamroga and van der Hoek, 2004, van der Hoek and Wooldridge, 2003, van Ditmarsch and Knight, 2014] including the latest one, called truly perfect recall (also referred as no-forgetting semantics) [Bulling et al., 2014], the coalition of g_1 and g_2 does not have such an ability to win, since they claim that coalitional abilities require general knowledge or even common knowledge.

Moreover, these semantic variants fail to preserve the coalition monotonicity which is a desirable property for coalitional abilities in Section 3.1, that is, if a coalition can achieve some goal, then its superset can achieve this goal as well. To demonstrate this idea, consider Figure 5.2 depicting a variant of the game in Example 5.1

by only switching the available actions of two guessers. i.e., let guesser g_1 choose the shell and guesser g_2 take no action. Then it is clear that guesser g_1 can win no matter what the others do, as she sees the location of the ball and can pick up the right shell. It should follow that as a coalition, the coalition of two guessers g_1 and g_2 can still win this game. However, according to most existing semantics, though guesser g_1 has the ability to win, this ability no longer holds once she forms a coalition with guesser g_2.

These counterintuitive phenomena motivate a variant of semantics for ATL based on the assumption of knowledge sharing within coalitions.

5.2 ATL with Knowledge Sharing

In this section, we provide a variant semantics for ATL based on the assumption of knowledge sharing in coalitions, and then investigate the properties of ATL under this semantics.

5.2.1 The Language

Let Φ be a finite set of atomic propositions and N be a nonempty finite set of agents. The language of ATL, denoted by $\mathcal{L}_{ATL}$, is defined by the following grammar:

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

where $p \in \Phi$ and $C \subseteq N$.

A coalition operator $\langle\langle C\rangle\rangle\varphi$ intuitively expresses that group C can cooperate to ensure that φ. The temporal operator $\Box$ means “from now on (always)” and other temporal operators are $\mathcal{U}$ (“until”) and $\bigcirc$ (“in the next state”). The dual operator

$\Diamond$ of $\Box$ ("either now or at some point in the future") is defined as $\Diamond\varphi =_{def} \neg\Box\neg\varphi$ or $\Diamond\varphi =_{def} \top\mathcal{U}\varphi$.

In particular, the standard epistemic operators [Fagin et al., 2003b] can be defined as follows:

$$\mathsf{K}_r\varphi =_{def} \langle\langle r\rangle\rangle\varphi\mathcal{U}\varphi \qquad \mathsf{D}_C\varphi =_{def} \langle\langle C\rangle\rangle\varphi\mathcal{U}\varphi$$
$$\widehat{\mathsf{K}}_r =_{def} \neg\mathsf{K}_r\neg\varphi \qquad \widehat{\mathsf{D}}_C\varphi =_{def} \neg\mathsf{D}_C\neg\varphi$$

As we will show in the semantics, these abbreviations capture their standard intuitions, i.e., "$\mathsf{K}_r\varphi$" says agent r knows φ, and $\mathsf{D}_C\varphi$ means it is distributed knowledge among coalition C that φ. The dual operators of K and D, denoted by $\widehat{\mathsf{K}}$ and $\widehat{\mathsf{D}}$, respectively, are defined as usual.

5.2.2 The Semantics

The semantics is built upon the *imperfect information concurrent game structure (iCGS)* [Schobbens, 2004, van der Hoek and Wooldridge, 2003].

Definition 5.1. An iCGS is a tuple $\mathcal{T} = (N, \Phi, W, \mathcal{A}, \{R_r\}_{r\in N}, d, \delta, \pi)$ where

- $N = \{r_1, \cdots, r_k\}$ is a nonempty finite set of *agents*.
- Φ is a finite set of *atomic propositions*.
- W is a nonempty finite set of *states*.
- $\mathcal{A}$ is a nonempty finite set of *actions*.
- $R_r \subseteq W \times W$ is an equivalence relation for agent r indicating the states that are indistinguishable from her viewpoint.
- $d : N \times W \to 2^{\mathcal{A}}\backslash\{\emptyset\}$ is a mapping specifying a nonempty set of available actions at each state for each agent with the requirement that every agent

always knows which actions are available to her, i.e., for all $w, w' \in W$ and $r \in N$, $d(r,w) = d(r,w')$ whenever wR_rw'. The set of joint actions at w for N is denoted as $D(w) = \Pi_{r\in N} d(r,w)$.

- $\delta : W \times D(W) \to W$ is the *transition function* specifying an outcome state $\delta(w,\alpha) \in W$ for each pair $(w,\alpha) \in W \times D(W)$.
- $\pi : \Phi \to 2^W$ is a standard *valuation function*.

A *computation* λ is an infinite sequence of states and actions $w_0 \overset{\alpha_1}{\to} w_1 \overset{\alpha_2}{\to} w_2 \cdots$, where for each $j \geq 1$, $\alpha_j \in D(w_{j-1})$ and $\delta(w_{j-1}, \alpha_j) = w_j$. Any finite segment of a computation is called a *history*. The set of all histories in $\mathcal{T}$ is denoted by H. We use $\lambda[j]$ to denote the j-th state of computation λ, $\lambda[j,k]$ $(0 \leq j \leq k)$ to denote the segment of λ from the j-th state to the k-th state, and $\lambda[j,\infty]$ to denote the subcomputation of λ starting from stage $j \geq 0$. The length of a history $h \in H$, denoted by $|h|$, is defined as the number of actions. Given $\alpha \in D(w)$, let $\alpha(r)$ denote agent r's individual action in joint action α.

The following definition specifies what an agent with perfect recall and perfect reasoning capabilities can in principle know at a special stage of an imperfect information game.

Definition 5.2. Two histories $h = w_0 \overset{\alpha_1}{\to} w_1 \overset{\alpha_2}{\to} \cdots \overset{\alpha_m}{\to} w_m$ and $h' = w'_0 \overset{\alpha'_1}{\to} w'_1 \overset{\alpha'_2}{\to} \cdots \overset{\alpha'_n}{\to} w'_n$ are equivalent for agent $r \in N$, denoted by $h \approx_r h'$, iff

1. $m = n$,
2. $w_j R_r w'_j$ for any $0 \leq j \leq m$, and
3. $\alpha_k(r) = \alpha'_k(r)$ for any $1 \leq k \leq m$.

In particular, two computations λ and λ' are equivalent up to stage $j \geq 0$ for agent $r \in N$, denoted by $\lambda \approx^j_r \lambda'$, iff $\lambda[0,j] \approx_r \lambda'[0,j]$.

Intuitively, two histories are indistinguishable for an agent if (1) they have the same length, (2) their corresponding states are indistinguishable for this agent, and (3) the agent takes the same action at each corresponding stage. This notion of perfect recall is more like the notion of perfect recall in [Thielscher, 2010, van Ditmarsch and Knight, 2014] as well as the notion of perfect recall in extensive games [Kuhn, 1953] by requiring that an agent remember the past states as well as her own actions. This is stronger than that in most epistemic ATL-style logics which use the state-based equivalence without taking the actions into consideration.

It should be noted that Ruan and Thielscher [2012] propose a way to embed actions to a state so that the state-based equivalence can achieve the same meaning; nevertheless, the advantages to explicitly add actions are at least twofold: on the one hand, it can deal with situations where different actions may lead to indistinguishable states. For instance, consider the following game structure with a single agent 1 in Figure 5.3. According to the state-based equivalence, the agent

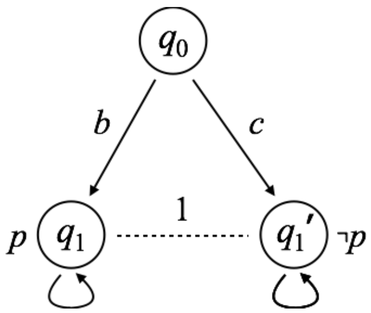

FIGURE 5.3: A game structure $\mathcal{T}$.

cannot distinguish the two histories $q_0 \xrightarrow{b} q_1$ and $q_0 \xrightarrow{c} q_1$, then at q_0 she does not know whether or not p holds next, but according our notion of perfect recall, the two histories are different to her, since she takes different actions at state q_0, then at q_0 she should know whether p holds or not at the next state; on the other hand, as we will see in the following sections, the introduction of actions plays an important role to prove the results about the interactions of coalition and epistemic operators.

As mentioned before, we assume that whenever a set of agents form a coalition to achieve their goals, these agents share their own knowledge before acting. To make this idea precise, we extend the indistinguishability relation of an agent to a group as the intersection of all its members' individual equivalence relation, i.e., $\approx_C^j = \bigcap_{r \in C} \approx_r^j$. Let $\approx_C^j (\lambda)$ denote the set of all computations that coalition C can not distinguish from λ up to stage j, i.e., $\approx_C^j (\lambda) = \{\lambda' \mid \lambda \approx_C^j \lambda'\}$.

A *strategy* is a plan of actions telling one agent what to do at each stage of a game. With knowledge sharing among members of a coalition, we say a strategy of agent $r \in C \subseteq N$ is *uniform* w.r.t C, if the strategy specifies the same action for r at all histories which are indistinguishable for coalition C.

Definition 5.3. Given $C \subseteq N$ and $r \in C$, *a uniform perfect recall strategy for agent r w.r.t C* is a function $f_r^C : H \to \mathcal{A}$ such that for all histories $h, h' \in H$,

1. if $h \approx_C h'$ then $f_r^C(h) = f_r^C(h')$, and
2. $f_r^C(h) \in d_r(last(h))$, where $last(h)$ denotes the last state of h.

Intuitively, a uniform perfect recall strategy for an agent in a group tells her a unique legal action to take at each history such that the actions for indistinguishable histories of the group are the same. In particular, the standard notion of uniform strategies with respect to individual knowledge can be viewed as a special case when C is a singleton. In the rest of the chapter, we simply write f_r^C for f_r when C is clear, and call a uniform perfect recall strategy *a strategy* for short.

A *joint strategy* for coalition $C \subseteq N$, denoted by F_C, is a vector of its members' individual strategies, i.e., $\langle f_r \rangle_{r \in C}$. Function $\mathcal{P}(f_r, h)$ returns the set of all computations that can occur when agent r's strategy f_r executes after an initial history h. Formally, $\lambda \in \mathcal{P}(f_r, h)$ iff $\lambda[0, |h|] = h$ and for any $j \geq |h|$, $f_r(\lambda[0, j]) = \theta_r(\lambda, j)$ where $\theta_r(\lambda, j)$ is the action of agent r taken at stage j on computation λ. Then the set of all computations complying with joint strategy

F_C after h, denoted by $\mathcal{P}(F_C, h)$, is defined as the intersection of its members' sets, i.e., $\mathcal{P}(F_C, h) = \bigcap_{r\in C} \mathcal{P}(f_r, h)$. This also indicates that if a group is characterized by full coordination both on the level of strategies and knowledge, we may view the group as a single agent whose abilities and knowledge are the sum of those of all the members [Kaźmierczak et al., 2014].

We are now in the position to provide a semantics for ATL based on the assumption of knowledge sharing in coalitions. Formulas are interpreted over triples consisting of a model, a computation and an index which indicates the current stage.

Definition 5.4. Let $\mathcal{T}$ be an iCGS. Given a computation λ in $\mathcal{T}$, a stage $j \in \mathbb{N}$ and a formula $\varphi \in \mathcal{L}_{ATL}$, φ is true (or satisfied) at j of λ under $\mathcal{T}$, denoted by $\mathcal{T}, \lambda, j \models \varphi$, according to the following definition:

$$
\begin{array}{lll}
\mathcal{T}, \lambda, j \models p & \text{iff} & p \in \pi(\lambda[j]) \\
\mathcal{T}, \lambda, j \models \neg\varphi & \text{iff} & \mathcal{T}, \lambda, j \not\models \varphi \\
\mathcal{T}, \lambda, j \models \varphi_1 \wedge \varphi_2 & \text{iff} & \mathcal{T}, \lambda, j \models \varphi_1 \text{ and } \mathcal{T}, \lambda, j \models \varphi_2 \\
\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle \bigcirc \varphi & \text{iff} & \exists F_C\ \forall \lambda' \approx^j_C \lambda\ \forall \lambda'' \in \mathcal{P}(F_C, \lambda'[0, j]) \\
 & & \mathcal{T}, \lambda'', j+1 \models \varphi \\
\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle \Box \varphi & \text{iff} & \exists F_C\ \forall \lambda' \approx^j_C \lambda\ \forall \lambda'' \in \mathcal{P}(F_C, \lambda'[0, j]) \\
 & & \forall k \geq j\ \mathcal{T}, \lambda'', k \models \varphi \\
\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle \varphi_1 \mathcal{U} \varphi_2 & \text{iff} & \exists F_C\ \forall \lambda' \approx^j_C \lambda \forall \lambda'' \in \mathcal{P}(F_C, \lambda'[0, j]) \\
 & & \exists k \geq j,\ \mathcal{T}, \lambda'', k \models \varphi_2, \text{ and} \\
 & & \forall j \leq t < k,\ \mathcal{T}, \lambda'', t \models \varphi_1
\end{array}
$$

The interpretation for coalition operator $\langle\langle C\rangle\rangle\varphi$ captures its precise meaning that coalition C by sharing its members' knowledge can cooperate to enforce that φ. Alternatively, the agents in C distributedly know that they can enforce that φ. A formula φ is *valid* in an iCGS $\mathcal{T}$, written $\mathcal{T} \models \varphi$, if $\mathcal{T}, \lambda, j \models \varphi$ for any computation $\lambda \in \mathcal{T}$ and any stage j. A formula φ is *valid*, denoted by $\models \varphi$, if it is valid in every iCGS $\mathcal{T}$.

The following proposition shows that the abbreviations for the epistemic operators capture their intended meanings, just as we expected.

Proposition 5.1. *Given an iCGS $\mathcal{T}$, a computation λ of $\mathcal{T}$ and a stage $j \in \mathbb{N}$,*

1. $\mathcal{T}, \lambda, j \models \mathsf{K}_r\varphi$ *iff for all* $\lambda' \approx_r^j \lambda$, $\mathcal{T}, \lambda', j \models \varphi$.

2. $\mathcal{T}, \lambda, j \models \mathsf{D}_C\varphi$ *iff for all* $\lambda' \approx_C^j \lambda$, $\mathcal{T}, \lambda', j \models \varphi$.

Proof. We only prove the first clause, and the second one is proved in a similar way. Let us show that $\mathcal{T}, \lambda, j \models \langle\langle r\rangle\rangle\varphi\mathcal{U}\varphi$ iff for all $\lambda' \approx_r^j \lambda$, $\mathcal{T}, \lambda', j \models \varphi$. Suppose $\mathcal{T}, \lambda, j \models \langle\langle r\rangle\rangle\varphi\mathcal{U}\varphi$ and for all $\lambda' \approx_r^j \lambda$, then there is f_r such that for any $\lambda'' \in \mathcal{P}(\lambda'[0,j], f_r)$, $\mathcal{T}, \lambda'', j \models \varphi$. And $\lambda'[0,j] = \lambda''[0,j]$, so $\mathcal{T}, \lambda', j \models \varphi$. The other direction is straightforward according to the truth condition for the coalition operator with $\mathcal{U}$. □

Let us go back to the shell game. We now show that this semantics justifies the intuition that the coalition of two guessers can cooperate to win by sharing knowledge.

Example 5.1 (*continued.*) *Consider the model $\mathcal{T}_1$ in Figure 5.1. It is easy to check that at stage 1 on the left computation $\lambda_1 := q_0q_1q_2\cdots$, neither guesser g_1 nor guesser g_2 has the ability to win at the next stage, i.e., $\mathcal{T}_1, \lambda_1, 1 \not\models \langle\langle g_1\rangle\rangle\bigcirc win$ and $\mathcal{T}_1, \lambda_1, 1 \not\models \langle\langle g_2\rangle\rangle\bigcirc win$. On the other hand, when g_1 and g_2 form a coalition, they can cooperate to win, as guesser g_2 is able to distinguish the history q_0q_1 from the history q_0q_1' after sharing knowledge. Thus, $\mathcal{T}_1, \lambda_1, 1 \models \langle\langle\{g_1, g_2\}\rangle\rangle\bigcirc win$.*

Regarding the coalition monotonicity, consider the model $\mathcal{T}_2$ in Figure 5.2. At stage 1 on the left computation $\lambda_1 := q_0q_1q_2\cdots$, guesser g_1 has the ability to win at the next stage by choosing the left shell, i.e., $\mathcal{T}_2, \lambda_1, 1 \models \langle\langle g_1\rangle\rangle\bigcirc win$. Furthermore, when g_1 and g_2 form a coalition, guesser g_1 still has the ability to make the coalition win, i.e., $\mathcal{T}_2, \lambda_1, 1 \models \langle\langle\{g_1, g_2\}\rangle\rangle\bigcirc win$.

It should be noted that the reason why alternative semantics fail to keep the coalition monotonicity property is that their interpretations of coalition operator $\langle\langle C\rangle\rangle\varphi$ are given without allowing knowledge sharing in coalitions. Specifically, the coalition operator is interpreted with respect to either the union of each member's equivalence relation or its transitive reflexive closure, which means the coalitional abilities implicitly require general knowledge or common knowledge. Neither general knowledge nor common knowledge is coalitionally monotonic. Different from them, distributed knowledge has the coalition monotonicity property.

5.3 Properties of Coalition Operators

We begin by showing that the coalitional abilities specified by coalition operators satisfy the plausible properties in Section 3.1

Proposition 5.2. *For all* C, $C' \subseteq N$ *and all* $\varphi, \psi \in \mathcal{L}_{ATL}$,

1. $\models \neg\langle\langle C\rangle\rangle \bigcirc \bot$
2. $\models \langle\langle C\rangle\rangle \bigcirc \top$
3. $\models \langle\langle C\rangle\rangle \bigcirc (\varphi \wedge \psi) \rightarrow \langle\langle C\rangle\rangle \bigcirc \varphi$
4. $\models \langle\langle C\rangle\rangle \bigcirc \varphi \rightarrow \langle\langle C'\rangle\rangle \bigcirc \varphi$ *where* $C \subseteq C'$
5. $\models \langle\langle C\rangle\rangle \bigcirc \varphi \wedge \langle\langle C'\rangle\rangle \bigcirc \psi \rightarrow \langle\langle C \cup C'\rangle\rangle \bigcirc (\varphi \wedge \psi)$ *where* $C \cap C' = \emptyset$
6. $\models \langle\langle C\rangle\rangle \bigcirc \varphi \rightarrow \neg\langle\langle N \backslash C\rangle\rangle \bigcirc \neg\varphi$
7. $\models \neg\langle\langle \emptyset\rangle\rangle \neg \bigcirc \varphi \rightarrow \langle\langle N\rangle\rangle \bigcirc \varphi$

Similarly for the $\Box$ *and* $\mathcal{U}$ *operators.*

Proof. Given an arbitrary iCGS $\mathcal{T}$, for any computation λ of $\mathcal{T}$ and any stage $j \in \mathbb{N}$, 1 follows from the fact that $\mathcal{T}, \lambda', j+1 \not\models \bot$ for any computation $\lambda' \approx_C^j \lambda$.

2 follows from the fact that $\mathcal{T}, \lambda', j+1 \models \top$ for any computation $\lambda' \approx_C^j \lambda$.

3 is straightforward by $\models \varphi \wedge \psi \to \varphi$.

4 Assume $\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle \bigcirc \varphi$, then there is $F_C = \langle f_r \rangle_{r \in C}$ such that for all $\lambda' \approx_C^j \lambda$, for all $\lambda'' \in \mathcal{P}(F_C, \lambda'[0,j])$, $\mathcal{T}, \lambda'', j+1 \models \varphi$. Let $F_{C'}$ be the same as F_C for any $r \in C$. Then for all $\lambda_1 \approx_{C'}^j \lambda$, for all $\lambda_2 \in \mathcal{P}(F_{C'}, \lambda_1[0,j])$, we have $\lambda_2 \in \mathcal{P}(F_C, \lambda'[0,j])$, so $\mathcal{T}, \lambda_2, j+1 \models \varphi$. Thus, $\mathcal{T}, \lambda, j \models \langle\langle C'\rangle\rangle \bigcirc \varphi$.

5 Assume $\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle \bigcirc \varphi \wedge \langle\langle C'\rangle\rangle \bigcirc \psi$, then there is $F_C^1 = \langle f_r^1 \rangle_{r \in C}$ such that for all $\lambda_1 \approx_C^j \lambda$, for all $\lambda_1' \in \mathcal{P}(F_C^1, \lambda_1[0,j])$, $\mathcal{T}, \lambda_1', j+1 \models \varphi$, and there is $F_{C'}^2 = \langle f_r^2 \rangle_{r \in C'}$ such that for all $\lambda_2 \approx_{C'}^j \lambda$, for all $\lambda_2' \in \mathcal{P}(F_{C'}^2, \lambda_2[0,j])$, $\mathcal{T}, \lambda_2', j+1 \models \psi$. We define the joint strategy $F_{C \cup C'} = \langle f_r \rangle_{r \in C \cup C'}$ as follows: for any history $h \in H$ and any $r \in C \cup C'$,

$$f_r(h) = \begin{cases} f_r^1(h) & \text{if } r \in C; \\ f_r^2(h) & \text{if } r \in C'. \end{cases}$$

This is well-defined by $C \cap C' = \emptyset$. It is easy to check that $F_{C \cup C'}$ is the joint strategy for coalition $C \cup C'$ to achieve both φ and ψ at the next state. So $\mathcal{T}, \lambda, j \models \langle\langle C \cup C'\rangle\rangle \bigcirc (\varphi \wedge \psi)$.

6 follows from 1 and 5.

7. Assume $\mathcal{T}, \lambda, j \models \neg \langle\langle \emptyset \rangle\rangle \neg \bigcirc \varphi$, then there is $\lambda' \in \mathcal{P}(F_\emptyset, \lambda[0,j])$, $\mathcal{T}, \lambda', j+1 \models \varphi$. We construct a joint coalition $F_N = \langle f_r \rangle_{r \in N}$ for the grand coalition in terms of λ' as follows: for any history $h \in H$ and any $r \in N$,

$$f_r(h) = \begin{cases} \theta_r(\lambda', j) & \text{if } h = \lambda'[0,i] \text{ for some } i \in \mathbb{N}; \\ a^r \in d_r(last(h)) & \text{otherwise.} \end{cases}$$

This is well defined as the proper initial segment of λ' is unique. Then $\mathcal{P}(F_N, \lambda[0,j]) = \{\lambda'\}$, so $M, \lambda, j \models \langle\langle N \rangle\rangle \bigcirc \varphi$. $\square$

Clause 1 says that no coalition C can enforce the falsity, and Clause 2 states every coalition C can enforce the truth. Clauses 3 and 4 specify the outcome-monotonicity and the coalition-monotonicity, respectively. Clause 5 is the super-additivity property specifying disjoint coalitions can combine their strategies to achieve more. Clause 6 is called C-regularity specifying that it is impossible for a coalition and its complementary set to enforce inconsistency. The last clause is N-maximality prescribing that the grand coalition can bring about something, if it is not impossible, i.e., there is a computation to achieve it.

The next proposition provides interesting validities about epistemic and coalition operators.

Proposition 5.3. *For any $C \subseteq N$ and all $\varphi, \psi \in \mathcal{L}_{ATL}$,*

1. $\models \langle\langle C\rangle\rangle \bigcirc \varphi \leftrightarrow \langle\langle C\rangle\rangle \bigcirc \mathsf{D}_C \varphi$
2. $\models \langle\langle C\rangle\rangle \bigcirc \varphi \leftrightarrow \mathsf{D}_C \langle\langle C\rangle\rangle \bigcirc \varphi$
3. $\models \langle\langle C\rangle\rangle \Box \varphi \leftrightarrow \langle\langle C\rangle\rangle \Box \mathsf{D}_C \varphi$
4. $\models \langle\langle C\rangle\rangle \Box \varphi \leftrightarrow \mathsf{D}_C \langle\langle C\rangle\rangle \Box \varphi$
5. $\models \langle\langle C\rangle\rangle \mathsf{D}_C \varphi \mathcal{U} \mathsf{D}_C \psi \rightarrow \langle\langle C\rangle\rangle \varphi \mathcal{U} \psi$
6. $\models \langle\langle C\rangle\rangle \varphi \mathcal{U} \psi \leftrightarrow \mathsf{D}_C \langle\langle C\rangle\rangle \varphi \mathcal{U} \psi$

Proof. We only give proof for the first two clauses, and the proof for the others is similar.

1. For every iCGS $\mathcal{T}$, every computation λ of $\mathcal{T}$ and every stage $j \in \mathbb{N}$, assume $\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle \bigcirc \varphi$, then there is $F_C = \langle f_r \rangle_{r \in C}$ such that for all $\lambda' \approx_C^j \lambda$, for all $\lambda'' \in \mathcal{P}(F_C, \lambda'[0, j])$, $\mathcal{T}, \lambda'', j+1 \models \varphi$. We next show that F_C is the joint strategy to verify $\langle\langle C\rangle\rangle \bigcirc \mathsf{D}_C \varphi$. Suppose not for a contradiction that there is $\lambda_1 \approx_C^j \lambda$, there is $\lambda_2 \in \mathcal{P}(F_C, \lambda_1[0, j])$, there is $\lambda_3 \approx_C^{j+1} \lambda_2$ such that $\mathcal{T}, \lambda_3, j+1 \not\models \varphi$, then $\lambda_3 \approx_C^j \lambda$. And $\theta_r(\lambda_3, j) = \theta_r(\lambda_2, j) = f_r(\lambda_2[0, j])$ for every $r \in C$, so there is some

$\lambda^* \in \bigcup_{\lambda' \approx^j_C \lambda} \mathcal{P}(F_C, \lambda'[0,j])$ such that $\lambda^*[0, j+1] = \lambda_3[0, j+1]$. By assumption we have $\mathcal{T}, \lambda^*, j+1 \models \varphi$, then $\mathcal{T}, \lambda_3, j+1 \models \varphi$: a contradiction. Thus, $\mathcal{T}, \lambda, j \models \langle\langle C \rangle\rangle \bigcirc \mathsf{D}_C \varphi$. The other direction is straightforward by $\models \mathsf{D}_C \varphi \to \varphi$.

2. For every iCGS $\mathcal{T}$, every computation λ of $\mathcal{T}$ and every stage $j \in \mathbb{N}$, assume $\mathcal{T}, \lambda, j \models \langle\langle C \rangle\rangle \bigcirc \varphi$, then there is $F_C = \langle f_r \rangle_{r \in C}$ such that for all $\lambda' \approx^j_C \lambda$, for all $\lambda'' \in \mathcal{P}(F_C, \lambda'[0,j])$, $\mathcal{T}, \lambda'', j+1 \models \varphi$. We next prove that for any $\lambda^* \approx^j_C \lambda$, $\mathcal{T}, \lambda^*, j \models \langle\langle C \rangle\rangle \bigcirc \varphi$. Let us take the strategy F_C. It is easy to check that for all $\lambda_1 \approx^j_C \lambda^*$, for all $\lambda_2 \in \mathcal{P}(F_C, \lambda_1[0,j])$, by $\approx^j_C (\lambda^*) = \approx^j_C (\lambda)$, we have $\mathcal{T}, \lambda_2, j+1 \models \varphi$. Thus, $\mathcal{T}, \lambda, j \models \mathsf{D}_C \langle\langle C \rangle\rangle \bigcirc \varphi$. The other direction is straightforward. $\square$

Note that it is not the case that $\models \langle\langle C \rangle\rangle \varphi \mathcal{U} \psi \to \langle\langle C \rangle\rangle \mathsf{D}_C \varphi \mathcal{U} \mathsf{D}_C \psi$. Here is a counter-example. Consider the model $\mathcal{T}_3$ in Figure 5.4 with two agents 1, 2, and states $\{q_0, q_1, q_1', q_2, q_2'\}$, where $q_1 R_1 q_1'$, but not for 2, and all the other states can be distinguished by both agents. There are two propositions p and q such that $\pi(p) = \{q_1\}$ and $\pi(q) = \{q_1', q_2\}$. The transitions are depicted as in the Figure.

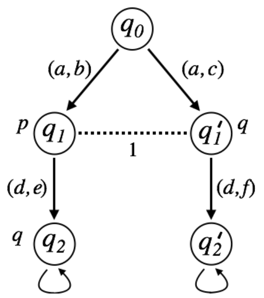

FIGURE 5.4: Counter-model $\mathcal{T}_3$.

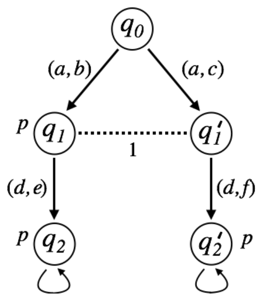

FIGURE 5.5: Counter-model $\mathcal{T}_4$.

Consider the left computation $\lambda := q_0 q_1 q_2 \cdots$. It is easy to check that $\mathcal{T}_3, \lambda, 1 \models \langle\langle 1 \rangle\rangle p \mathcal{U} q$, but $\mathcal{T}_3, \lambda, 1 \not\models \langle\langle 1 \rangle\rangle \mathsf{K}_1 p \mathcal{U} \mathsf{K}_1 q$.

It follows from Proposition 6.4 that the distributed knowledge operator and the coalition operator are interchangeable w.r.t temporal operators $\bigcirc$ and $\Box$.

Corollary 5.4. *For any $C \subseteq N$ and any $\varphi \in \mathcal{L}_{ATL}$,*

1. $\models \langle\langle C \rangle\rangle \bigcirc \mathsf{D}_C \varphi \leftrightarrow \mathsf{D}_C \langle\langle C \rangle\rangle \bigcirc \varphi$

2. $\models \langle\langle C\rangle\rangle\Box \mathsf{D}_C\varphi \leftrightarrow \mathsf{D}_C\langle\langle C\rangle\rangle\Box\varphi$

5.4 Knowledge Sharing and Coalitional Abilities

In this section, we investigate the interplay between knowledge shared by a group of agents and its coalitional abilities in ATL with the proposed semantics. To start with, we show that, similar to [Belardinelli, 2014, 2015], the standard fixed-point characterizations of coalition operators for ATL [Goranko and van Drimmelen, 2006] fail under this semantics.

Proposition 5.5. *For any $C \subseteq N$ and all $\varphi, \psi \in \mathcal{L}_{ATL}$,*

1. $\not\models \varphi \wedge \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Box\varphi \to \langle\langle C\rangle\rangle\Box\varphi$
2. $\not\models \varphi \vee \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Diamond\varphi \to \langle\langle C\rangle\rangle\Diamond\varphi$
3. $\not\models \langle\langle C\rangle\rangle\Diamond\varphi \to \varphi \vee \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Diamond\varphi$
4. $\not\models \psi \vee (\varphi \wedge \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\varphi\mathcal{U}\psi) \to \langle\langle C\rangle\rangle\varphi\mathcal{U}\psi$
5. $\not\models \langle\langle C\rangle\rangle\varphi\mathcal{U}\psi \to \psi \vee (\varphi \wedge \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\varphi\mathcal{U}\psi)$

Here is a counter-example for the first one. Consider the model $\mathcal{T}_4$ in Figure 5.5, which is obtained from $\mathcal{T}_3$ by only changing the valuations. There is one proposition p such that $\pi(p) = \{q_1, q_2, q_2'\}$. Consider $\varphi := p$ and the left computation $\lambda := q_0q_1q_2\cdots$. Then it is easy to check that $\mathcal{T}_4, \lambda, 1 \models p$, $\mathcal{T}_4, \lambda, 1 \models \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle\Box p$, but $\mathcal{T}_4, \lambda, 1 \not\models \langle\langle 1\rangle\rangle\Box p$. Thus, $\mathcal{T}_4, \lambda, 1 \not\models p \wedge \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle\Box p \to \langle\langle 1\rangle\rangle\Box p$.

On the other hand, the next proposition shows that the converse direction for $\Box$ holds under the proposed semantics.

Proposition 5.6. *For any $C \subseteq N$ and any $\varphi \in \mathcal{L}_{ATL}$,*

$$\models \langle\langle C\rangle\rangle\Box\varphi \to \varphi \wedge \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Box\varphi.$$

Proof. For every iCGS $\mathcal{T}$, every computation λ of $\mathcal{T}$ and every stage $j \in \mathbb{N}$, assume $\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle\Box\varphi$, then there is $F_C = \langle f_r\rangle_{r\in C}$ such that for all $\lambda' \approx^j_C \lambda$, for all $\lambda'' \in \mathcal{P}(F_C, \lambda'[0, j])$, $\mathcal{T}, \lambda'', k \models \varphi$ for all $k \geq j$. In particular, $\lambda \approx^j_C \lambda$. Then for all $\lambda^* \in \mathcal{P}(F_C, \lambda[0, j])$, $\mathcal{T}, \lambda^*, j \models \varphi$. And by $\lambda[0, j] = \lambda^*[0, j]$, we obtain $\mathcal{T}, \lambda, j \models \varphi$.

We next prove that $\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Box\varphi$. It suffices to show that F_C is just the joint strategy to verify the both coalition operators. That is, for all $\lambda_1 \approx^j_C \lambda$, for all $\lambda_2 \in \mathcal{P}(F_C, \lambda_1[0, j])$, for all $\lambda_3 \approx^{j+1}_C \lambda_2$, for all $\lambda_4 \in \mathcal{P}(F_C, \lambda_3[0, j + 1])$, we want to prove that $\mathcal{T}, \lambda_4, i \models \varphi$ for all $i \geq j + 1$. By $\lambda_4 \in \mathcal{P}(F_C, \lambda_3[0, j + 1])$, we have $\lambda_4[0, j + 1] = \lambda_3[0, j + 1]$, then $\lambda_4 \approx^{j+1}_C \lambda_2$, so $\lambda_4 \approx^j_C \lambda_2$ and $\lambda_4 \in \bigcup_{\lambda_1\approx^j_C\lambda} \mathcal{P}(F_C, \lambda_1[0, j])$. And by the assumption, we have $\mathcal{T}, \lambda_4, i \models \varphi$, so $\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Box\varphi$.

Thus, $\mathcal{T}, \lambda, j \models \varphi \wedge \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Box\varphi$. $\Box$

We now present the main result about the interactions of distributed knowledge and coalitional abilities in the context of imperfect information. Recall that $\widehat{\mathsf{K}}$ and $\widehat{\mathsf{D}}$ are the dual operators of K and D, respectively.

Theorem 5.7. *For any $C \subseteq N$ and for all $\varphi, \psi \in \mathcal{L}_{ATL}$,*

1. $\models \langle\langle C\rangle\rangle\Box\varphi \leftrightarrow \mathsf{D}_C\varphi \wedge \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Box\varphi$
2. $\models \langle\langle C\rangle\rangle\Diamond\varphi \rightarrow \widehat{\mathsf{D}}_C\varphi \vee \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Diamond\varphi$
3. $\models \mathsf{D}_C\varphi \vee \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Diamond\varphi \rightarrow \langle\langle C\rangle\rangle\Diamond\varphi$
4. $\models \langle\langle C\rangle\rangle\varphi\mathcal{U}\psi \rightarrow \widehat{\mathsf{D}}_C\psi \vee (\mathsf{D}_C\varphi \wedge \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\varphi\mathcal{U}\psi)$
5. $\models \mathsf{D}_C\psi \vee (\mathsf{D}_C\varphi \wedge \langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\varphi\mathcal{U}\psi) \rightarrow \langle\langle C\rangle\rangle\varphi\mathcal{U}\psi$

Clause 1 says that a coalition by sharing their knowledge can cooperate to maintain φ if, and only if the coalition distributedly knows φ at the current stage, and there is a joint strategy for this coalition to possess this ability at the next stage. Clause

2 states that a coalition by sharing their knowledge can eventually achieve φ, only if either the coalition considers it is possible that φ at the current stage, or it has a joint strategy to possess this ability at the next stage. Clause 3 provides a sufficient condition that a coalition by sharing their knowledge can eventually achieve φ, if either it is distributed knowledge among the coalition that φ holds, or the coalition can cooperate to achieve this ability at the next stage. The intuitions behind the last two clauses are similar. We next provide the proof.

Proof. 1. For every iCGS $\mathcal{T}$, every computation λ of $\mathcal{T}$ and every stage $j \in \mathbb{N}$, assume $\mathcal{T}, \lambda, j \models \mathsf{D}_C\varphi \wedge \langle\langle C\rangle\rangle \bigcirc \langle\langle C\rangle\rangle \Box\varphi$, then $\mathcal{T}, \lambda, j \models \mathsf{D}_C\varphi$ and $\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle \bigcirc \langle\langle C\rangle\rangle \Box\varphi$. By the latter, we obatin that there is $F_C^1 = \langle f_r^1\rangle_{r\in C}$ such that for all $\lambda_1 \approx_C^j \lambda$, for all $\lambda_2 \in \mathcal{P}(F_C^1, \lambda_1[0,j])$, $\mathcal{T}, \lambda_2, j+1 \models \langle\langle C\rangle\rangle \Box\varphi$, then there is $F_C^{2\cdot x} = \langle f_r^{2\cdot x}\rangle_{r\in C}$, where $x = \lambda_2[0, j+1]$, such that for all $\lambda_3 \approx_C^{j+1} \lambda_2$, for all $\lambda_4 \in \mathcal{P}(F_C^{2\cdot x}, \lambda_3[0, j+1])$, $\mathcal{T}, \lambda_4, t \models \varphi$ for all $t \geq j+1$. We next construct a new joint strategy $F_C = \langle f_r\rangle_{r\in C}$ based on F_C^1 and $F_C^{2\cdot x}$. To this end, we need the following notation[1]. Let

$$X = \{\lambda'[0,j] \xrightarrow{\alpha} w \mid \lambda' \approx_C^j \lambda, \forall r \in C(\alpha(r) = f_r^1(\lambda'[0,j])), \text{ and } w = \delta(\lambda'[j], \alpha)\}.$$

Intuitively, X is the set of all possible outcomes generated by the agents in C taking the next actions specified by F_C^1 from a history that is indistinguishable from history $\lambda[0,j]$. With this notation, we now define strategy $F_C = \langle f_r\rangle_{r\in C}$ as follows: For all $h \in H(\mathcal{T})$ and for all $r \in C$,

$$f_r(h) = \begin{cases} f_r^{2\cdot l}(h) & \text{if } \exists l \in X \text{ such that } l \text{ is a segment of } h; \\ f_r^1(h) & \text{otherwise.} \end{cases}$$

Note that this is well-defined, because if a history h has a segment in X, there is only one such segment because according to the definition of equivalence relations over histories all histories in X has the same length.

[1]It should be noted that this idea is borrowed from [van Ditmarsch and Knight, 2014].

We now show that F_C is just the joint strategy we need to verify $\langle\langle C\rangle\rangle\Box\varphi$. That is, for all $\lambda' \approx_C^j \lambda$, for all $\lambda'' \in \mathcal{P}(F_C, \lambda'[0,j])$, we want to prove that for all $s \geq j$, $\mathcal{T}, \lambda'', s \models \varphi$. By $|l| > |\lambda'[0,j]|$ for any $l \in X$, we have that there is no $l \in X$ such that l is a segment of $\lambda'[0,j]$, then by the definition of F_C, $F_C(\lambda'[0,j]) = F_C^1(\lambda'[0,j])$, so $\lambda''[0,j+1] \in X$ and $\mathcal{T}, \lambda'', j+1 \models \langle\langle C\rangle\rangle\Box\varphi$. From the latter, we obtain that for all $\lambda^\bullet \approx_C^{j+1} \lambda''$, for all $\lambda^* \in \mathcal{P}(F_C^{2\cdot y}, \lambda^\bullet[0,j+1])$, where $y = \lambda''[0,j+1]$, $\mathcal{T}, \lambda^*, t \models \varphi$ for all $t \geq j+1$. And by $\lambda''[0,j+1] \in X$, we have $\lambda^\bullet[0,j+1] \in X$. Then by the definition of F_C and the assumption $\lambda'' \in \mathcal{P}(F_C, \lambda'[0,j])$, we obtain $\lambda'' \in \mathcal{P}(F_C^{2\cdot y}, \lambda^\bullet[0,j+1])$, so for all $s \geq j+1$, $\mathcal{T}, \lambda'', s \models \varphi$. And by the assumption $\mathcal{T}, \lambda, j \models \mathsf{D}_C\varphi$, we have $\mathcal{T}, \lambda'', j \models \varphi$. Thus, $\mathcal{T}, \lambda'', s \models \varphi$ for all $s \geq j$, so $\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle\Box\varphi$.

The other direction is proved by a similar method in Proposition 5.6.

2. Assume $\mathcal{T}, \lambda, j \models \langle\langle C\rangle\rangle\Diamond\varphi$, then there is $F_C = \langle f_r\rangle_{r\in C}$ such that for all $\lambda' \approx_C^j \lambda$, for all $\lambda'' \in \mathcal{P}(F_C, \lambda'[0,j])$, $\mathcal{T}, \lambda'', k \models \varphi$ for some $k \geq j$. Further assume $\mathcal{T}, \lambda, j \models \mathsf{D}_C\neg\varphi$, then for all $\lambda^* \approx_C^j \lambda$, $\mathcal{T}, \lambda^*, j \models \neg\varphi$, so for all $\lambda' \approx_C^j \lambda$, for all $\lambda'' \in \mathcal{P}(F_C, \lambda'[0,j])$, there is $k > j$ such that $\mathcal{T}, \lambda'', k \models \varphi$. It is not hard to show that F_C is just the joint strategy to verify $\langle\langle C\rangle\rangle\bigcirc\langle\langle C\rangle\rangle\Diamond\varphi$.

Clause 3 and Clause 5 are proved by a similar method of Clause 1, and Clause 4 is proved by a similar method of Clause 2. $\Box$

It should be noted that with knowledge sharing, the fixed-point characterization only holds for the coalition operator with box $\langle\langle C\rangle\rangle\Box$. Here are two simple counter-examples for the other two cases.

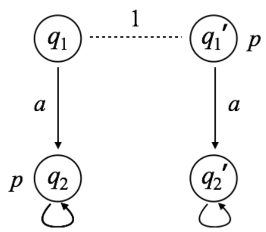

FIGURE 5.6: Counter-model $\mathcal{T}_5$.

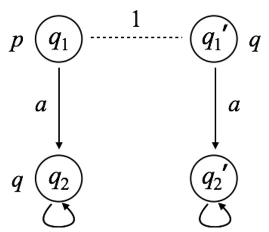

FIGURE 5.7: Counter-model $\mathcal{T}_6$.

Consider the counter-model $\mathcal{T}_5$ in Figure 5.6 with one single agent 1 and states $\{q_1, q_1', q_2.q_2'\}$ where $q_1 R_1 q_1'$. There is one single atomic proposition p such that $\pi(p) = \{q_1', q_2\}$. The transitions are depicted in the figure and the reflexive loops are omitted. Consider the computation $\lambda := q_1 q_2 \cdots$. Then $\mathcal{T}_5, \lambda, 1 \models \langle\langle 1 \rangle\rangle \Diamond p$, but $\mathcal{T}_5, \lambda, 1 \not\models \mathsf{K}_1 \varphi \vee \langle\langle 1 \rangle\rangle \bigcirc \langle\langle 1 \rangle\rangle \Diamond p$. $\mathcal{T}_6$ is obtained by changing the valuations. There are two atomic propositions p and q such that $\pi(p) = \{q_1\}$ and $\pi(q) = \{q_1', q_2\}$. Consider the computation $\lambda := q_1 q_2 \cdots$. Then $\mathcal{T}_6, \lambda, 1 \models \langle\langle 1 \rangle\rangle p \mathcal{U} q$, but $\mathcal{T}_6, \lambda, 1 \not\models \mathsf{K}_1 q \vee (\mathsf{K}_1 \varphi \wedge \langle\langle 1 \rangle\rangle \bigcirc \langle\langle 1 \rangle\rangle p \mathcal{U} q)$.

In particular, we have the following result for a single agent.

Corollary 5.8. *For any $r \in N$ and all $\varphi, \psi \in \mathcal{L}_{ATL}$,*

1. $\models \langle\langle r \rangle\rangle \Box \varphi \leftrightarrow \mathsf{K}_r \varphi \wedge \langle\langle r \rangle\rangle \bigcirc \langle\langle r \rangle\rangle \Box \varphi$
2. $\models \langle\langle r \rangle\rangle \Diamond \varphi \rightarrow \widehat{\mathsf{K}}_r \varphi \vee \langle\langle r \rangle\rangle \bigcirc \langle\langle r \rangle\rangle \Diamond \varphi$
3. $\models \mathsf{K}_r \varphi \vee \langle\langle r \rangle\rangle \bigcirc \langle\langle r \rangle\rangle \Diamond \varphi \rightarrow \langle\langle r \rangle\rangle \Diamond \varphi$
4. $\models \langle\langle r \rangle\rangle \varphi \mathcal{U} \psi \rightarrow \widehat{\mathsf{K}}_r \psi \vee (\mathsf{K}_r \varphi \wedge \langle\langle r \rangle\rangle \bigcirc \langle\langle r \rangle\rangle \varphi \mathcal{U} \psi)$
5. $\models \mathsf{K}_r \psi \vee (\mathsf{K}_r \varphi \wedge \langle\langle r \rangle\rangle \bigcirc \langle\langle r \rangle\rangle \varphi \mathcal{U} \psi) \rightarrow \langle\langle r \rangle\rangle \varphi \mathcal{U} \psi$

5.5 Discussion and Summary

Reasoning about coalitional abilities and strategic interactions is fundamental in analysis of games and multi-agent systems. In recent years, many logical formalisms have been proposed for this purpose. For a latest survey of this topic, please refer to [Herzig, 2015, van Ditmarsch et al., 2015] . In this following, we review the literature which is most related to our work.

In the context of imperfect information, several semantic variants have been proposed for ATL based on different interpretations of agents' abilities [Ågotnes et al., 2007, Alechina et al., 2016, Jamroga and van der Hoek, 2004, Schobbens, 2004, van der Hoek and Wooldridge, 2003]. In particular, Jamroga and Bulling [2014, 2011] provide formal comparisons of validity sets for semantic variants of ATL, and Dima et al. [2015] point the subtle expressiveness properties of ATL with imperfect information and perfect recall. Similar to the no forgetting semantics of Bulling et al. [2014], our semantics is also history-based w.r.t a computation and an index, but there are fundamental differences between their work and ours. Firstly, we consider a finer notion of perfect recall by taking both past states and actions into consideration so as to deal with situations where different actions may have the same effects. Secondly, our notion of joint strategies is defined in terms of distributed knowledge instead of general knowledge, as we assume that when a set of agents form a coalition, they are able to share their knowledge before cooperating to ensure a goal.

A few epistemic-ATL style logics have been proposed to investigate the interaction of group knowledge and coalitional abilities [Bulling and Jamroga, 2014, Guelev et al., 2011, Huang et al., 2016, Schobbens, 2004, van Ditmarsch and Knight, 2014]. Among them, Guelev et al. [2011] present a variant of ATL with knowledge, perfect recall and past. Different from our motivation, they use distributed knowledge of coalitions to have a decidable model-checking problem, since Dima and Tiplea [2011] prove that model checking for ATL under imperfect information and perfect recall is undecidable. Moreover, to facilitate the formulation of epistemic goals, they also allow the unrestricted use of the past connectives which are not involved in our work. van Ditmarsch and Knight [2014] propose three types of coalition operators to specify different cases of how all agents in the coalition cooperate to enforce a goal. Among them, the communication strategy operator $\langle\langle C\rangle\rangle_c$ captures the intuition behind our coalition operator. Specifically, we have the following correspondence.

Given an iCGS $\mathcal{T}$, a computation λ of $\mathcal{T}$ and a stage $j \in \mathbb{N}$, let φ be any formula of the form $\bigcirc\psi$, $\Box\psi$ or $\psi_1 \mathcal{U} \psi_2$. Then

$$\mathcal{T}, \lambda, j \models \langle\langle C \rangle\rangle \varphi \text{ iff } \mathcal{T}, \lambda[0, j] \models_{euATL} \langle\langle C \rangle\rangle_c \varphi.$$

Their work is different from ours in the following aspects: firstly, they propose two epistemic versions of ATL, namely uATL and euATL, to address the issue of uniformity of strategies in the combination of strategic and epistemic systems, while we introduce a variant of semantics without adding epistemic operators to explore the interplay of epistemic and coalition operators; secondly, their results mainly focus on the relations and logical properties of three coalition operators, and the interplay of distributed knowledge and coalition operators is not involved in their work; thirdly, their meaning of coalition is more subtle than ours. Besides the communication strategy operator, there are *the active coalitional strategy operator* and *the passive coalitional strategy operator* in their work. The comparison with these two strategy operators is less straightforward, since they are based on assumptions of coalitions without sharing knowledge.

It is also worth mentioning that Herzig and Troquard [2006] adopt a similar meaning of coalition so as to capture the notion of "knowing how to play". Besides the different motivations, that work is based on STIT framework and only considers one-step uniform strategies without investigating the interplay of epistemic and coalition operators. The notion of uniform strategies and the relevant complexity have been extensively investigated in [Bozzelli et al., 2015, Maubert and Pinchinat, 2013, 2014, Maubert et al., 2013]. Among them, Maubert and Pinchinat [2014] propose a formula language to specify a general notion of uniform strategies and study an automated procedure to synthesize strategies subject to a uniformity property. In particular, their definition subsumes our notion of uniform strategies.

In summary, this chapter has proposed a variant of semantics for ATL with imperfect information and perfect recall so as to investigate the interplay of the knowledge shared by a group of agents and its coalitional abilities. We have showed that this semantics can not only preserve the plausible properties of coalitional abilities, but also provide a finer notion of perfect recall requiring an agent remembers the past states as well as the past actions. More importantly, we have investigated the interplay of epistemic and coalition operators, which can be seen as an attempt towards the question: which kind of group knowledge is required for a group to achieve some goal in the context of imperfect information [Herzig, 2015]. We did not investigate the computational complexity of ATL with this semantics; nevertheless, Guelev et al. [2011] show a decidable model-checking algorithm for a variant of ATL with knowledge, perfect recall and past based on the assumption that strategies are uniform with respect to distributed knowledge of the coalition. We expect a similar decidability result holds under our setting.

Chapter 6

Reason-Based Collective Choice

This chapter changes the focus from (joint) actions to (collective) preferences, and use the approach to combine actions via their priorities in Chapter 2 to deal with the collective decision-making. It proposes a modal logic by extending propositional logic with the prioritized connective $\triangledown$ introduced in Section 2.2.2 for modelling individual and collective choices in social choice theory.

6.1 Background

Social Choice Theory deals with the problem of how to aggregate individual preferences into a social or collective preference so as to reach a collective decision [Gaertner, 2009]. In the simplest setting when individual preferences are given by orderings over available alternatives, social choice is to select a rule that maps a profile of individual preference orderings into a social preference ordering over the same alternatives, and then make a collective choice from the alternatives in terms of the social preference ordering. However, in many situations, individual preferences may not be given in the form of an ordering over alternatives but the "reasons" that lead to the preference. For instance, when buying a property, we may express our preference in the way of specifying which locations we like the house to be, how many bedrooms we want it to have and which price range we can afford. These reasons not only induce a preference ordering over the set of alternatives, say a ranking over the houses available on the market, but also convey more information than the preference ordering, which is crucial for understanding rational choice of an agent, i.e., *reason-based choice* [Clippel and Eliaz, 2012, Dietrich and List, 2013b, 2016]. A reason-based choice of an agent is a choice based on reasons and the agent acts on the basis of reasons [Dietrich and List, 2013b].

Representing reasons for a preference is more fundamental than representing the preference itself. There are a number of ways that the concept of a reason can be formalized. A simple way is to express a reason as a propositional formula [Lang, 2004]. For instance, if we want a four-bedroom house located in Mountain View, the reason can be represented as $Mountain\_View \wedge Four\_bedroom$. However, most often not all of the reasons can be satisfied, and we have to make some sort of compromises. One of the most natural and convenient methods is to sort reasons. If the best option is unavailable, the agent may be then satisfied by the second best option (or third, *etc*). For instance, we might want to express that we want to buy a four-bedroom house located in Mountain View most; if it is impossible, a four-bedroom house located in Menlo Park is also fine. Here is the place where the prioritized connective $\triangledown$ introduced in Section 2.2.2 comes into play. For example, our reasons for house choice with the above-mentioned priority over locations can be naturally represented as $(Mountain\_View \wedge Four\_bedroom) \triangledown (Menlo\_Park \wedge Four\_bedroom)$.

Logical representation of reason-based choices does not provide a solution to the problem of reason-based social choice. A first idea would be that we convert the reason-based preference of each agent into a preference ordering over alternatives, and then apply a conventional preference aggregation rule to deduce the social preference ordering. Unfortunately, this does not provide a solution to the problem, because the outcome of the social preference is no longer reason-based, which does not give reasons for collective choice.

This chapter aims to address this problem by proposing a modal logic for modelling individual and collective choices based on reasons. We first extend the language of propositional logic with the prioritized connective $\triangledown$ so that each formula in this language can express not only reasons for choices (i.e., properties of alternatives) but also priorities over the reasons. Each formula of this logic determines a preference ordering over alternatives on the basis of priorities over the reasons. In such a way, the problem of collective choice is reduced to how to aggregate a set of formulas into a single formula. We then define a few plausible collective choice rules within the same language, which are specified by Arrow's conditions [Gaertner, 2009]. Interestingly, all Arrowian conditions are plausible under this setting except Independence of Irrelevant Alternatives (IIA). This gives us a natural way to circumvent Arrow's impossibility result [Gaertner, 2009]. We also show a possibility result by replacing IIA with Monotonicity [Fagin et al., 2003a]. Finally we develop a model-checking algorithm for this logic so as to automatically generate individual and collective choices, which is rarely achieved by the existing logics for social choice theory [Endriss, 2011].

6.2 Reason-Based Choice Logic

In this section, we establish a modal logic for modelling reason-based choices. In the rest of the chapter, we call this logic *reason-based choice logic*, denoted as RCL for short.

6.2.1 The Language

The language of RCL is obtained by extending propositional logic with the prioritized connective $\triangledown$ introduced in Section 2.2.2.

Definition 6.1. The language, denoted by $\mathcal{L}_{RCL}$, consists of:

- a nonempty finite set Φ_0 of atomic propositions;
- propositional connectives $\neg$ and $\wedge$;
- a binary modality $\triangledown$, representing priority over reasons instead of actions in Chapter 2.

A formula φ in $\mathcal{L}_{RCL}$ is generated by the following BNF:

$$\varphi ::= A \mid \varphi \triangledown \varphi$$

where A is a standard propositional formula built as follows:

$$A ::= p \mid \neg A \mid A \wedge A$$

where $p \in \Phi_0$.

The formulas are in two levels. The formulas in the lower level are the standard propositional formulas, used for describing reasons or properties of alternatives. The other logical connectives $\vee$, $\rightarrow$, $\leftrightarrow$ and the logical constants $\top$, $\bot$ are introduced in the standard way. The formulas in the upper level are designed to express priorities over reasons. A formula $\varphi_1 \triangledown \varphi_2$ means "choose an alternative to make φ_1 true; if no alternatives make it true, choose one to make φ_2 true". In other words, the agent gives a higher priority to the reason specified by φ_1 than to that specified by φ_2 in decision making.

It should be noted that the prioritized connective $\triangledown$ is from Section 2.2.2. We use it to give priorities over reasons instead of actions. Here we do not allow the prioritized connective being nested in any propositional connective. The reason is that once this is allowed, the intuition behind prioritized choice will be lost. For instance, it is unclear what the prior reasons are determined by the conjunction of two prioritized choices $\varphi_1 \triangledown \psi_1$ and $\varphi_2 \triangledown \psi_2$. There are many ways to merge prioritized choices. In fact, this is exactly a problem of choice aggregation, which is the main theme of this chapter.

As the BNF shows, nesting prioritized choice formulas is allowed. For example, $(\varphi_1 \triangledown \varphi_2) \triangledown \varphi_3$ is a well-formed formula in $\mathcal{L}_{RCL}$. As we will show in next section, $\triangledown$ is associative, so the following abbreviation becomes meaningful:

$$\varphi_1 \triangledown \varphi_2 \triangledown \cdots \triangledown \varphi_m =_{def} ((\varphi_1 \triangledown \varphi_2) \triangledown \cdots) \triangledown \varphi_m$$

6.2.2 The Semantics

The semantics for the classical propositional logic interprets each atomic proposition with a truth value either true or false. Formally, an interpretation I is a function that maps Φ_0 to $\{\mathbf{true}, \mathbf{false}\}$. Alternatively, an interpretation can also be expressed as a set of literals[1], in which the positive literals represent the atomic propositions that are true under the interpretation, while the negative literals represent the atomic propositions which are false.

Let W be the set of alternatives from which an agent has to choose. Assume that each alternative is uniquely specified by its properties/attributes/characters expressed by atomic propositions. Take the restaurant menu as an example. Assume a restaurant offers a number of dishes $W = \{x, y, z, ...\}$ for us to choose. Each dish is characterised by its ingredients and styles. For instance, *Bouillabaisse Royale* is a French dish made up of fish fillets, prawns, scallops, scampi, mussels and cooked in a fish-and-tomato stock. If each character is expressed by an atomic proposition in $\Phi_0 = \{p, q, r, ...\}$, a dish can be uniquely identified by an interpretation of Φ_0. In general, we can simply view each alternative in W as an interpretation over Φ_0, if we represent each property/attribute/character of the alternatives by an atomic proposition in Φ_0. Therefore, W becomes a set of interpretations over Φ_0, i.e., $W \subseteq 2^{\Phi_0}$. In the following, we assume any set W of

[1] A literal is an atomic proposition or its negation.

alternatives is non-empty. Note that W is finite and does not have to contain all interpretations of Φ_0.

Next we consider how to interpret a propositional formula. Suppose that you want to eat seafood in a restaurant. You will then choose a dish from the restaurant menu that contains seafood, such as fish, prawn or others. Assume that, as we mentioned above, we represent the characters of food in atomic propositions. Then the statement $seafood$ can be expressed in a propositional formula, such as $(fish \vee prawn) \wedge lemon$. You choose the dishes (represented as interpretations of the language) that can satisfy this formula. In general, an alternative w in W is a candidate of our choice if it satisfies our reasons for selection represented by A, i.e., $w \models A$.

Finally, we consider the interpretation of a prioritized formula. As we mentioned above, $\varphi \triangledown \psi$ means *"choose an alternative to meet φ; if it is impossible, choose one to meet ψ"*. Consider an alternative $w \in W$, if w satisfies φ, it certainly satisfies $\varphi \triangledown \psi$. However, if none of the alternatives in W satisfies φ, then an alternative w' satisfies $\varphi \triangledown \psi$ only if w' satisfies ψ.

Based on above intuitive discussion, we are now ready to define the truth conditions for all formulas in our language.

Definition 6.2. Given a set, $W \subseteq 2^{\Phi_0}$, of alternatives and for any $w \in W$, a formula $\varphi \in \mathcal{L}_{RCL}$ is true (or satisfied) at w in W, denoted by $W, w \models \varphi$, iff

$$\begin{array}{lll}
W, w \models p & \text{iff} & p \in w \\
W, w \models \neg A & \text{iff} & w \not\models A \\
W, w \models A \wedge B & \text{iff} & w \models A \text{ and } w \models B \\
W, w \models \varphi_1 \triangledown \varphi_2 & \text{iff} & W, w \models \varphi_1, \text{ or } (W, w \models \varphi_2 \text{ and } W, w' \not\models \varphi_1 \text{ for all } w' \in W).
\end{array}$$

We say φ is *valid* in W, denoted by $W \models \varphi$, if $W, w \models \varphi$ for every $w \in W$; φ is *valid*, denoted by $\models \varphi$, if $W \models \varphi$ for any $W \subseteq 2^{\Phi_0}$. Given any two formulas φ, $\psi \in \mathcal{L}_{RCL}$, φ is a logical consequence of ψ, denoted by $\varphi \models \psi$, iff for any W and any $w \in W$, if $W, w \models \varphi$, then $W, w \models \psi$.

The following result shows that the prioritized connective is associative.

Proposition 6.1. *For all φ_1, φ_2 and $\varphi_3 \in \mathcal{L}_{RCL}$, for any W and any $w \in W$,*

$$W, w \models \varphi_1 \triangledown (\varphi_2 \triangledown \varphi_3) \textit{ iff } W, w \models (\varphi_1 \triangledown \varphi_2) \triangledown \varphi_3$$

Proof.

$$\begin{array}{ll}
\text{Assume} & W, w \models \varphi_1 \triangledown (\varphi_2 \triangledown \varphi_3) \\
\text{iff} & W, w \models \varphi_1, \text{ or (for all } v \in W, W, v \not\models \varphi_1 \text{ and } W, w \models \varphi_2 \triangledown \varphi_3) \\
\text{iff} & W, w \models \varphi_1, \text{ or (for all } v \in W, W, v \not\models \varphi_1 \text{ and } (W, w \models \varphi_2, \\
 & \text{ or (for all } v \in W, W, v \not\models \varphi_2 \text{ and } W, w \models \varphi_3))) \\
\text{iff} & W, w \models \varphi_1, \text{ or (for all } v \in W, W, v \not\models \varphi_1 \text{ and } W, w \models \varphi_2), \\
 & \text{ or (for all } v \in W, W, v \not\models \varphi_1 \vee \varphi_2 \text{ and } W, w \models \varphi_3) \\
\text{iff} & W, w \models \varphi_1, \text{ or (for all } v \in W, W, v \not\models \varphi_1 \text{ and } W, w \models \varphi_2), \\
 & \text{ or (for all } v \in W (W, v \not\models \varphi_1, \text{ and (there is } u \in W \\
 & W, u \models \varphi_1 \text{ or } W, v \not\models \varphi_2)) \text{ and } W, w \models \varphi_3) \\
\text{iff} & W, w \models \varphi_1 \triangledown \varphi_2, \text{ or (for all } v \in W, W, v \not\models \varphi_1 \triangledown \varphi_2 \text{ and } W, w \models \varphi_3) \\
\text{iff} & W, w \models (\varphi_1 \triangledown \varphi_2) \triangledown \varphi_3
\end{array}$$

Thus, $W, w \models \varphi_1 \triangledown (\varphi_2 \triangledown \varphi_3)$ iff $W, w \models (\varphi_1 \triangledown \varphi_2) \triangledown \varphi_3$. □

6.2.3 Expressiveness and Succinctness

In this subsection, we investigate the expressiveness and succinctness of the language for preference representation.

6.2.3.1 Expressivity

Let us begin by how to extend priorities over reasons to a preference ordering over alternatives. *A preference ordering over alternatives is a reflexive, total and* *transitive relation over alternatives.* Hereafter, we show that the proposed language is simple, yet expressive enough to describe any preference ordering over alternatives.

Basically, we generate a preference ordering over W in terms of a formula φ, denoted by $\preceq_\varphi$, in this way: an alternative w *is preferred to* an alternative w', denoted by $w \preceq_\varphi w'$, if the maximal important reason that w satisfies is given at least a similar priority as the maximal important reason that w' satisfies.

Definition 6.3. Given $\varphi \in \mathcal{L}_{RCL}$ of the form $A_1 \triangledown A_2 \triangledown \cdots \triangledown A_m$ and an alternative $w \in W$, let

$$h(w) = \begin{cases} Min\{i \ : \ w \models A_i\} & \text{if } w \models \bigvee_{i=1}^{m} A_i; \\ \\ m+1 & \text{otherwise.} \end{cases}$$

Then for any two alternatives $w, w' \in W$, $w \ \preceq_\varphi \ w'$ iff $h(w) \ \leq \ h(w')$.

As usual, $w \prec_\varphi w' =_{def} w \preceq_\varphi w'$ and not $(w' \preceq_\varphi w)$, and $w \sim_\varphi w' =_{def} w \preceq_\varphi w'$ and $w' \preceq_\varphi w$. It follows that an alternative w with $h(w) = m + 1$ must fail to satisfy φ itself, i.e., $w \notin C(W, \varphi)$, since it dissatisfies all the reasons. We next use a simple example to illustrate how this works.

Example 6.1. *Given* $\Phi_0 = \{p, q, r, s\}$ *and* $\varphi = (p \wedge q) \triangledown (r \to s)$*, let* $W = \{w_1, w_2, w_3, w_4\}$*, where* $w_1 = \{p, q, r, \neg s\}$*,* $w_2 = \{p, q, r, s\}$*,* $w_3 = \{\neg p, q, r, \neg s\}$ *and* $w_4 = \{p, \neg q, r, s\}$*. Then we have* $h(w_1) = h(w_2) = 1$*,* $h(w_3) = 3$ *and* $h(w_4) = 2$*. Hence,* $w_1 \sim_\varphi w_2 \prec_\varphi w_4 \prec_\varphi w_3$*.*

The following proposition says that the language $\mathcal{L}_{RCL}$ can not only generate a preference ordering over W, but also express any preference ordering over W.

Proposition 6.2. *Given a set* W *of alternatives,*

1. *for any formula* $\varphi \in \mathcal{L}_{RCL}$*,* $\preceq_\varphi$ *is a preference ordering over* W*.*

2. *for any preference ordering $\preceq$ over W, there is a formula $\varphi \in \mathcal{L}_{RCL}$ such that $\preceq \; = \; \preceq_{\varphi}$.*

Proof. The first clause follows from the fact that $\preceq_{\varphi}$ is reflexive, total and transitive according to Definition 6.3.

For the other one, since W is distinguishable, each alternative w can be uniquely characterized by a propositional formula Γ_w. We construct a formula by composing propositional formulas with the prioritized connective according to the given preference ordering such that: $\Gamma_w \bigtriangledown \Gamma_{w'}$ if $w \prec w'$, and $\Gamma_w \vee \Gamma_{w'}$ if $w \sim w'$. □

6.2.3.2 Succinctness

We next investigate another property of this language: the relative space efficiency. Specifically, we show that one of the standard and compact languages, called the goal-based language $R_{GB}^{bestout}$ [Benferhat et al., 1993], can be translated to our language in polynomial-size, and vice versa.

Definition 6.4. A goal base GB is a tuple $\langle \{G_1, \cdots, G_n\}, r \rangle$ where

- $\{G_1, \cdots, G_n\}$ is a finite set of propositional formulas;
- r is an associated function from $\mathbb{N}$ to $\mathbb{N}$.

If $r(i) = j$, then j is called the rank of formula G_i. By convention, a lower rank means a higher priority. The priority on goals can be extended to a preference ordering over alternatives via *best-out ordering*.

Definition 6.5. Let $r_{GB}(w) = Min\{r(i) \mid w \not\models G_i\}$. Then for any two alternatives $w, w' \in W$, $w \preceq_{GB}^{bo} w'$ iff $r_{GB}(w) \geq r_{GB}(w')$.

The following result says that our language has the same degree of succinctness with $R_{GB}^{bestout}$.

Proposition 6.3. *There is a polynominal-size translation from $\mathcal{L}_{RCL}$ to $R^{bestout}_{GB}$, and vice versa.*

Proof. The polynominal-size translation Tr_1 from $\mathcal{L}_{RCL}$ to $R^{bestout}_{GB}$ is given as follows: for any formula φ of the form $A_1 \triangledown \cdots \triangledown A_m$, $Tr_1(\varphi) = \langle\{G_1, \cdots, G_n\}, r\rangle$ such that $m = n$, $G_i = \bigvee_{k=1}^{m-i+1} A_k$ for any $1 \leq i \leq m$, and $r(i) = i$ for any $1 \leq i \leq m$.

Conversely, the polynominal-size translation Tr_2 from $R^{bestout}_{GB}$ to $\mathcal{L}_{RCL}$ is given as follows: for any base goal $\langle\{G_1, \cdots, G_n\}, r\rangle$, there is a priority relation over $\{G_1, \cdots, G_n\}$ according to their ranks given by r. Without loss of generalization, suppose r is a permutation, and $\{G'_1, \cdots, G'_n\}$ is the prioritized goals generated by r, then $Tr_2(\langle\{G_1, \cdots, G_n\}, r\rangle) = A_1 \triangledown \cdots \triangledown A_m$ such that $m = n$ and $A_i = \bigwedge_{k=1}^{m-i+1} G'_k$ for any $1 \leq i \leq m$.

It is not hard to show that for any two alternatives $x, y \in W$, $x \preceq_\varphi y$ iff $x \preceq^{bo}_{Tr_1(\varphi)} y$, and $x \preceq^{bo}_{GB} y$ iff $x \preceq^{bo}_{Tr_2(GB)} y$. $\square$

It follows that RCL and $R^{bestout}_{GB}$ have the same space efficiency for preference representation, and thus the succinctness results of $R^{bestout}_{GB}$ in [Coste-Marquis et al., 2004] hold for RCL as well.

6.3 Reason-Based Collective Choice

In this section, we first show how to use RCL to represent a choice, then extend this logic to the multi-agent case for collective choice, and finally define a set of plausible collective choice rules specified by Arrow's conditions.

6.3.1 Choice Set

A choice set is a subset of alternatives that are selected by reasons specified by a formula $\varphi \in \mathcal{L}_{RCL}$. We now introduce the syntactical representation of a choice set.

Definition 6.6. Given a set W of alternatives and a formula $\varphi \in \mathcal{L}_{RCL}$, *the choice set* specified by φ in W, denoted by $C(W, \varphi)$, is defined as follows:

$$C(W, \varphi) = \{w \in W : W, w \models \varphi\}$$

Intuitively, choice set $C(W, \varphi)$ includes all the alternatives in W that satisfy φ. To illustrate how to use prioritized connectives for making a choice, let us consider the following example.

Example 6.2. *Three friends Ann, Kate and Bill are going to watch a movie together. Ann is a super fan of cartoon comedies, therefore is eager to find one. If nothing, other comedies are ok, and a fiction as the least option. Kate also likes cartoons but only non-fiction cartoons. If nothing, she picks a comedy, and a fiction if nothing else. Finally, Bill will surely go for a fiction, and a non-cartoon is also ok. If nothing, any movie seems fine for him. They find three movies are on show: $Gravity$, $Flipped$ and $Frozen$. It is known that:*

- *$Gravity$ is a fiction but not a comedy or cartoon;*
- *$Flipped$ is a comedy but not a fiction or cartoon;*
- *$Frozen$ is a cartoon comedy but not a fiction.*

Let us first formalize this example. The set of atomic properties is $\Phi_0 = \{Fiction$, $Comedy$, $Cartoon\}$. The set of feasible alternatives is $W = \{Gravity$, $Flipped$, $Frozen\}$, where

$$\begin{aligned} Gravity &= \{Fiction, \neg Comedy, \neg Cartoon\}; \\ Flipped &= \{\neg Fiction, \neg Cartoon, Comedy\}; \\ Frozen &= \{\neg Fiction, Comedy, Cartoon\}. \end{aligned}$$

The prioritized reasons of Ann, Kate and Bill are expressed by RCL-formulas as follows:

$$\begin{aligned} \varphi_{Ann} &= (Comedy \wedge Cartoon) \triangledown Comedy \triangledown Fiction \\ \varphi_{Kate} &= (Cartoon \wedge \neg Fiction) \triangledown Comedy \triangledown Fiction \\ \varphi_{Bill} &= Fiction \triangledown \neg Cartoon \triangledown \top \end{aligned}$$

According to their individual reasons, Ann and Kate both choose the movie $Frozen$, and Bill chooses the movie Gravity. This intuitive judgment is validated by the model, i.e., $W, Frozen \models \varphi_{Ann}$, $W, Frozen \models \varphi_{Kate}$, and $W, Gravity \models \varphi_{Bill}$. Then it follows that

- $C(W, \varphi_{Ann}) = \{Frozen\}$ *(that is, $Frozen$ is chosen based on Ann's reason.)*
- $C(W, \varphi_{Kate}) = \{Frozen\}$ *(that is, $Frozen$ is chosen based on Kate's reason.)*
- $C(W, \varphi_{Bill}) = \{Gravity\}$ *(that is, $Frozen$ is chosen based on Bill's reason.)*

Since they would like to watch a movie together, so a natural question arises: which movie should they choose? We will deal with this question in the next section.

Before handling the collective dimension, we now show that the approach to use prioritized connectives for making choices is *rational*, that is, this approach satisfies the two standard rationality conditions: *the contraction condition* and *the expansion condition* [Gaertner, 2009].

Contraction Condition. *Given two sets of alternatives W, W' with $W \subseteq W'$, for all $w \in W$ and for all $\varphi \in \mathcal{L}_{RCL}$, if $w \in C(W', \varphi)$, then $w \in C(W, \varphi)$.*

This condition requires that if you choose some alternative from a set of alternatives, and if this alternative remains available in a more restricted set, then you also choose it from the restricted one.

Expansion Condition. *Given two sets of alternatives W, W' with $W \subseteq W'$, for all alternatives $w, w' \in W$ and for any formula $\varphi \in \mathcal{L}_{RCL}$, if ($w \in C(W, \varphi)$ and $w' \in C(W, \varphi)$), then ($w \in C(W', \varphi)$ iff $w' \in C(W', \varphi)$).* This condition requires that if you choose two alternatives from a set of alternatives, then you choose them or not choose them at the same time from a lager set.

Proposition 6.4. *For any set W of alternatives and any formula $\varphi \in \mathcal{L}_{RCL}$, the choice set $C(W, \varphi)$ satisfies the contraction condition and the expansion condition.*

Proof. We prove this by induction on the structure of φ. It is straightforward for the case when φ is a propositional formula. We only show the case when $\varphi := \varphi_1 \triangledown \varphi_2$.

(Contraction Condition). The following claim is used for the main proof. It is easily proved by induction on the structure of φ.

Claim 6.5. For any $\varphi \in \mathcal{L}_{RCL}$ and W, W' with $W \subseteq W'$, if $W' \models \varphi$, then $W \models \varphi$.

With this claim, assume $w \in C(W', \varphi_1 \triangledown \varphi_2)$, then $W', w \models \varphi_1 \triangledown \varphi_2$, so $W', w \models \varphi_1$, or (for all $v \in W', W', v \not\models \varphi_1$ and $W', w \models \varphi_2$). And by induction hypothesis and Claim 6.5, $W, w \models \varphi_1$, or (for all $v \in W, W, v \not\models \varphi_1$ and $W, w \models \varphi_2$), we obtain $W, w \models \varphi_1 \triangledown \varphi_2$. Thus, $w \in C(W, \varphi_1 \triangledown \varphi_2)$.

(Expansion Condition). Assume $w \in C(W, \varphi_1 \triangledown \varphi_2)$ and $w' \in C(W, \varphi_1 \triangledown \varphi_2)$, then $W, w \models \varphi_1 \triangledown \varphi_2$ and $W, w' \models \varphi_1 \triangledown \varphi_2$. Further suppose not for a contradiction that for some $W' \supseteq W$ such that $w \in C(W', \varphi_1 \triangledown \varphi_2)$ and $w' \notin C(W', \varphi_1 \triangledown \varphi_2)$. Then $W', w \models \varphi_1 \triangledown \varphi_2$ and $W', w' \not\models \varphi_1 \triangledown \varphi_2$. That is,
(i) $W', w \models \varphi_1$, or (ii) (for all $v \in W', W', v \not\models \varphi_1$ and $W', w \models \varphi_2$), and
(iii) ($W', w' \not\models \varphi_1$ and there is $u \in W', W', u \models \varphi_1$), or ($iv$) ($W', w' \not\models \varphi_1$ and

$W', w' \not\models \varphi_2$).

We next consider four cases. Thanks to the associative property. It suffices to show the basic case when φ_1, φ_2 are propositional formulas.

If (i) and (iii), then $W, w \models \varphi_1$ and $W, w' \not\models \varphi_1$, so $W, w' \not\models \varphi_1 \triangledown \varphi_2$, contradicting with the assumption;

If (i) and (iv), it is prove in a similar way;

If (ii) and (iii): a contradiction;

If (ii) and (iv), then $W, w' \not\models \varphi_1$ and $W, w' \not\models \varphi_2$, so $W, w' \not\models \varphi_1 \triangledown \varphi_2$, contradicting with the assumption.

Thus, if $w \in C(W', \varphi_1 \triangledown \varphi_2)$, then $w' \in C(W', \varphi_1 \triangledown \varphi_2)$. The other case is proved in a similar way. □

6.3.2 The Multi-Agent Setting

Let us consider a finite set of agents $N = \{1, 2, \cdots, n\}$. Each agent $i \in N$ has her own reasons which are specified by a formula $\varphi_i \in \mathcal{L}_{RCL}$ of the form $A_1^i \triangledown A_2^i \triangledown \cdots \triangledown A_m^i$ such that

(*Individual Completeness*) $W \models \bigvee_{k=1}^{m} A_k^i$.

Individual completeness requires that each individual should take all the alternatives in W into consideration, that is, her priority over these reasons induces a rank for every alternative in W, which corresponds to the completeness requirement in preference aggregation [Gaertner, 2009]. This requirement guarantees that each individual has a non-empty choice set, i.e., $C(W, \varphi_i) \neq \emptyset$. In the following, we call a formula an *individual choice*, if it satisfies individual completeness. Given each individual choice φ_i, the vector $\langle \varphi_i \rangle_{i \in N}$ is called *a profile (of individual choices)*.

Finally, a collective choice rule is a function F that assigns to each profile $\langle \varphi_i \rangle_{i \in N}$ a single formula $\varphi \in \mathcal{L}_{RCL}$ of the form $A_1 \triangledown A_2 \triangledown \cdots \triangledown A_m$ such that

(*Collective Completeness*) $W \models \bigvee_{k=1}^{m} A_k$.

This condition guarantees that the aggregate formula should always determinate a collective alternative, i.e., $C(W, \varphi) \neq \emptyset$, and induce a collective preference ordering over alternatives based on priorities over reasons. The set of the profiles of individual choices is called the *domain* of F, denoted by $Dom(F)$.

Similar to preference aggregation, a collective choice rule must behave in a rational way and few constraints or conditions have to be set for enforcing this rationality. The very first requirement is that logically equivalent formulas should determine the same most preferable alternatives. However, the collective choice is actually not completely dependent on the set of individual most preferable alternatives. Most of the time individuals have to make some compromises, and the second, the third, or even the last most preferred alternatives might be taken into consideration. Since the standard notions of logical consequence and equivalence are defined based on the most preferable alternatives, so they need to be strengthened so as to handle collective choice.

Definition 6.7. Given a set W of alternatives, let $\varphi = A_1 \triangledown \cdots \triangledown A_m$ and $\psi = B_1 \triangledown \cdots \triangledown B_n$. Then ψ is a strong consequence of φ w.r.t W, denoted by $\varphi \Vdash_W \psi$, iff

1. $m \geq n$, and
2. $W \models A_k \to \bigvee_{i=1}^{k} B_i$ for any $1 \leq k \leq n$.

The intuitions behind this notion is the following: firstly, the length of the priority over reasons specified by φ is at least as large as that by ψ. This means that the number of ranks of alternatives generated by φ is at least as large as that given by ψ (Condition 1); secondly, for any alternative in W, its rank given by ψ is at least as high as the one given by φ (Condition 2). In particular, when $k = 1$, the most important reason in φ implies the most important reason in ψ. It follows that the most preferred alternative of φ (if exists) is also a most preferred alternative of ψ. To illustrate this idea, let us consider a simple example.

Example 6.3. *Consider three alternatives* $W = \{x, y, z\}$*, where* $x = \{p, \neg q, \neg r\}$*,* $y = \{q, \neg p, \neg r\}$ *and* $z = \{r, \neg q, \neg p\}$*. Let* $\varphi = p \triangledown q \triangledown r$ *and* $\psi = (p \vee r) \triangledown q$*. Then* $\varphi \Vdash_W \psi$ *because* $W \models p \rightarrow (p \vee r)$ *and* $W \models q \rightarrow (p \vee r \vee q)$*. Figure 6.1 illustrates the relation between the ranks for each alternative (the higher, the better).*

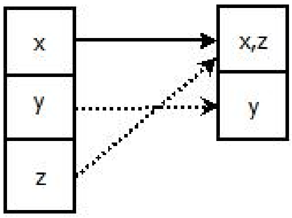

FIGURE 6.1: ψ (right) is a strong consequence of φ (left) w.r.t. W.

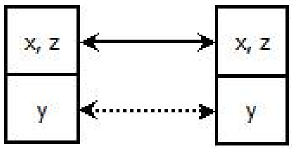

FIGURE 6.2: ψ (left) is strongly equivalent to χ (right) w.r.t. W.

Furthermore, the strong notion of equivalence is defined as follows:

Definition 6.8. Given a set W of alternatives and for all $\varphi, \psi \in \mathcal{L}_{RCL}$, φ is strongly equivalent to ψ w.r.t W, denoted by $\Vdash_W \varphi \equiv \psi$, iff $\varphi \Vdash_W \psi$ and $\psi \Vdash_W \varphi$.

This definition says that two strongly equivalent formulas have the same priority over reasons. In other words, they assign the same rank to every alternatives in W. Let us continue Example 6.3. Consider another formula $\chi = \neg q \triangledown q$, then $\Vdash_W \psi \equiv \chi$, as $W \models \neg q \leftrightarrow (p \vee r)$ and $W \models q \leftrightarrow \neg(p \vee r) \wedge q$. Figure 6.2 illustrates their strong equivalence.

The following proposition restates the strong equivalence in terms of the standard logical equivalence. In fact, it also serves as an important lemma for proving our main result, e.g., Theorem 6.8.

Proposition 6.6. *Given a set* W *of alternatives, let* $\varphi = A_1 \triangledown \cdots \triangledown A_m$ *and* $\psi = B_1 \triangledown \cdots \triangledown B_n$*. Then* $\Vdash_W \varphi \equiv \psi$ *iff*

1. $m = n$*,*
2. $W \models A_1 \leftrightarrow B_1$*, and*

3. $W \models (\bigwedge_{i=1}^{k-1} \neg A_i \wedge A_k) \leftrightarrow (\bigwedge_{i=1}^{k-1} \neg B_i \wedge B_k)$ *for any* $2 \leq k \leq n$*.*

Proof. We first show the direction from left to right. Assume $\Vdash_W \varphi \equiv \psi$, then $\varphi \Vdash_W \psi$ and $\psi \Vdash_W \varphi$. By Definition 6.7, the first and second conditions hold directly. We now show the third condition by induction on k.

- For $k = 2$, assume $W \models \neg A_1 \wedge A_2$, then for any $w \in W$, $W, w \models \neg A_1 \wedge A_2$, so $W, w \models \neg A_1$ and $W, w \models A_2$. And by Definition 6.7, we obtain that $W, w \models \neg B_1$ and $W, w \models B_1 \vee B_2$, so $W, w \models \neg B_1 \wedge B_2$. The other direction is proved in a similar way.

- For $k+1$, assume $W \models \bigwedge_{i=1}^{k} \neg A_i \wedge A_{k+1}$, then for any $w \in W$, $W, w \models \bigwedge_{i=1}^{k} \neg A_i$ and $W, w \models A_{k+1}$, so $W, w \models \bigwedge_{i=1}^{k} \neg B_i$, otherwise $W, w \models \bigvee_{i=1}^{k} B_i$, then there is some $1 \leq j \leq k$ such that $W, w \models B_j$, then by Definition 6.7, $W, w \models \bigvee_{i=1}^{j} A_i$, so $W, w \models \bigvee_{i=1}^{k} A_i$, contradicting with $W, w \models \bigwedge_{i=1}^{k} \neg A_i$. By $W, w \models A_{k+1}$ and Definition 6.7, we have $W, w \models \bigvee_{i=1}^{k+1} B_i$, so $W, w \models B_{k+1}$. Thus, $W \models \bigwedge_{i=1}^{k} \neg B_i \wedge B_{k+1}$. The other direction is proved in a similar way.

We next show the other direction from right to left. We need prove $\varphi \Vdash_W \psi$ and $\psi \Vdash_W \varphi$. Since they are symmetric, we only show $\varphi \Vdash_W \psi$, and the other one is proved in a similar way. According to Definition 6.7, it suffices to show

1. $m \geq n$, and

2. $W \models A_k \rightarrow \bigvee_{i=1}^{k} B_i$ for any $1 \leq k \leq n$.

The first condition holds directly. We now prove the second condition by induction on k. Assume for any $1 \leq t < k$, the result holds. Further suppose not for a contradiction that $W \not\models A_k \rightarrow \bigvee_{i=1}^{k} B_i$, then there is some $w \in W$ such that

$W, w \models A_k$ and $W, w \not\models \bigvee_{i=1}^{k} B_i$, so $W, w \models \bigwedge_{i=1}^{k} \neg B_i$. And by the assumption we get $W, w \not\models \bigwedge_{i=1}^{k-1} \neg A_i \wedge A_k$, then $W, w \not\models \bigwedge_{i=1}^{k-1} \neg A_i$, so $W, w \models \bigvee_{i=1}^{k-1} A_i$. Thus, there is some $1 \leq j \leq k-1$ such that $W, w \models A_j$. And by the induction hypothesis, we get $W, w \models \bigvee_{i=1}^{j} B_i$, then $W, w \models \bigvee_{i=1}^{k} B_i$: a contradiction. Thus, $W \models A_k \rightarrow \bigvee_{i=1}^{k} B_i$. $\square$

It should be noted that the strong consequence (equivalence) and the standard consequence (equivalence) are the same, if φ is a propositional formula. However, once φ is a formula with prioritized connectives, the two versions become different. As we will see in the next section, the strong version is designed to specify conditions for collective choice rules.

6.3.2.1 Conditions for Collective Choice Rules

We now investigate the conditions which a collective choice rule is expected to satisfy. Let F be a collective choice rule.

Unrestricted domain (U). *For any profile $\langle \varphi_i \rangle_{i \in N}$ of individual choices, $\langle \varphi_i \rangle_{i \in N} \in Dom(F)$.* The domain of the collective choice rule F includes all profiles of individual choices.

Anonymity (A). *For any profile $\langle \varphi_i \rangle_{i \in N}$ and any permutation $\sigma : N \rightarrow N$, $\Vdash_W F(\langle \varphi_i \rangle_{i \in N}) \equiv F(\langle \varphi_{\sigma(i)} \rangle_{i \in N})$.* This requires that the ordering among the agents should not affect the collective result, and the collective rule should treat each individual neutrally.

Monotonicity (M). *For any two profiles $\Phi = \langle \varphi_i \rangle_{i \in N}$, $\Phi' = \langle \varphi'_i \rangle_{i \in N}$ and for all $i \in N$, if $\varphi_i \Vdash_W \varphi'_i$, then $F(\Phi) \Vdash_W F(\Phi')$.* This condition is the qualitative counterpart of the monotonicity condition in [Fagin et al., 2003a]. It specifies that if for each individual the rank of an alternative in one profile is at least as high as that in the other, then the collective rank of the former is at least as high as that of the latter.

Pareto principle (P). *For any profile* $\Phi = \langle\varphi_i\rangle_{i\in N}$ *and for any formula* $\varphi \in \mathcal{L}_{RCL}$*, if* $\Vdash_W \varphi_i \equiv \varphi$ *for any* $i \in N$*, then* $\Vdash_W F(\Phi) \equiv \varphi$*.* Intuitively, if all individual priority over reasons are the same, then this condition requires that the collective priority over reasons should be the same as each individual's.

The following proposition says Non-dictatorship can be derived from Anonymity.

Proposition 6.7. *Every collective choice rule* F *satisfying* ***A*** *is non-dictatorial, i.e., there is no* $i \in N$ *such that* $\Vdash_W \varphi_i \equiv F(\Phi)$ *for any profile* $\Phi \in Dom(F)$*.*

Proof. Suppose not for a contradiction that there is a collective choice rule F satisfying **A** such that some $a \in N$ is a dictator for F. Consider a profile $\Phi = \langle\varphi_i\rangle_{i\in N}$, where $\not\Vdash_W \varphi_a \equiv \varphi_j$ for some $j \neq a$. We next construct a new profile $\Phi' = \langle\varphi'_i\rangle_{i\in N}$ as follows: Let φ'_a be φ_j, φ'_j be φ_a, and φ'_i be φ_i for all $i \in N\backslash\{a, j\}$. According to the dictator a, $\Vdash_W F(\Phi) \equiv \varphi_a$ and $\Vdash_W F(\Phi') \equiv \varphi'_a$. Since $\not\Vdash_W \varphi_a \equiv \varphi_j$, so $\not\Vdash_W F(\Phi) \equiv F(\Phi')$. But Φ' is a permutation of Φ, and by Condition **A**, we have $\Vdash_W F(\Phi) \equiv F(\Phi')$: a contradiction. □

Under this reason-based setting Universal domain, Non-dictatorship and Pareto principle correspond to their counterparts of Arrowian conditions in preference aggregation except Independence of Irrelevant alternatives which is replaced by Monotonicity. In the next section we will show that there are collective choice rules satisfying these conditions.

6.3.2.2 Collective Choice Rules

Besides representing choices and individual preferences, RCL is also able to express collective choice rules.

Let $\Phi = \langle\varphi_i\rangle_{i\in N}$ be a profile of individual choices where φ_i denotes the formula

$$A_1^i \triangledown A_2^i \triangledown \cdots \triangledown A_{m_i}^i.$$

Without loss of generalization, assume $m_1 < m_2 < \cdots < m_n$. We begin with a naive rule F_{grd} called *the grounded rule*[2] defined as follows:

Definition 6.9.

$$F_{grd}(\Phi) = \bigvee_{i=1}^{n} A_1^i \triangledown \bigvee_{i=1}^{n} A_2^i \triangledown \cdots \triangledown \bigvee_{i=1}^{n} A_{m_1}^i.$$

The intuition behind the grounded rule is that *the collective choice should be a best choice for at least one of the agents.* The grounded rule works in this way: first check whether there is any alternative satisfying one of individual's most important reasons, if there is such an alternative, simply choose this alternative; otherwise, go on and check whether there is any alternative satisfying one of the individual's second most important reasons. Continue this procedure until an alternative is found. This rule is simple and easy to be executed. However, it has drawbacks: it naively selects one of the individual's most preferred alternatives to be the collective choice without taking into account the ranks of this alternative in the others' preference orderings. Consider Example 6.2 again. According to F_{grd}, movie *Gravity* is a possible output for collective choice as *Bill* likes it most. However, this is counter-intuition since both Ann and Kate like *Gravity* the least. Hence, the worst-off dimension should be more considered.

The next rule F_{max} called *the maximal rule* improves this aspect. It is defined as follows:

Definition 6.10.

$$\begin{aligned} F_{max}(\Phi) = \; & \big(\bigwedge_{i=1}^{n} A_1^i\big) \triangledown \cdots \triangledown \big(\bigwedge_{i=1}^{n} (\bigvee_{j=1}^{m_1} A_j^i)\big) \triangledown \big(\bigwedge_{i=2}^{n} (\bigvee_{j=1}^{m_1+1} A_j^i)\big) \triangledown \cdots \\ & \triangledown \big(\bigwedge_{i=2}^{n} (\bigvee_{j=1}^{m_2} A_j^i)\big) \triangledown \cdots \triangledown \big(\bigwedge_{i=n-1}^{n} (\bigvee_{j=1}^{m_{n-1}} A_j^i)\big) \triangledown \big(\bigvee_{j=1}^{m_n} A_j^n\big) \end{aligned}$$

The idea behind the maximal rule F_{max} is to *maximize the situation of the worst-off.* This rule guarantees that an alternative has to be collectively selected, only if

[2]The notion 'groundedness' is borrowed from [Porello and Endriss, 2011].

its lowest rank in all individual preference orderings is the highest among the lowest ranks of all the other alternatives. This specification rules out the possibility of movie *Gravity* to be a collective choice in Example 6.2. The maximal rule proceeds as follows:

1. Check whether or not there is some alternative satisfying the most important reasons of all individuals (the conjunction of the first prioritized disjunct of each formula). If there is such an alternative, choose that alternative and halt; otherwise, go to the next step.
2. Check whether or not there is some alternative satisfying either the first or the second important reasons of all individuals (the conjunction of all the disjunctions of the first two prioritized disjuncts of each individual formula). If there is such an alternative, choose that alternative and halt; otherwise, go to the next step.
3. Continue above procedure until there is k such that some alternative satisfies the disjunction of first k prioritized disjuncts of all individual formulas.

Since each individual formula is complete and its length is finite, so k must exist. Note that the last line take care of the case that individual formulas may have different lengths. This rule is the qualitative counterpart of *Fagin's Algorithm* in database system, which is an efficient data aggregation algorithm with elegant mathematical properties [Fagin, 1996].

Finally, we define a family of *uniform quota rules* [Dietrich and List, 2007b]. Given a threshold $\tau \in \{1, 2, \cdots, n\}$, the corresponding uniform quota rule, denoted by F_τ, is defined as follows:

Definition 6.11.

$$
\begin{aligned}
F_\tau(\Phi) = \; & \Big(\bigvee_{C\subseteq N,|C|=\tau}\bigwedge_{i\in C} A_1^i\Big)\triangledown\cdots\triangledown\Big(\bigvee_{C\subseteq N,|C|=\tau}\bigwedge_{i\in C}(\bigvee_{j=1}^{m_1} A_j^i)\Big)\\
& \Big(\bigvee_{C\subseteq N\setminus\{1\},|C|=\tau}\bigwedge_{i\in C}(\bigvee_{j=1}^{m_1+1} A_j^i)\Big)\triangledown\cdots\triangledown\Big(\bigvee_{C\subseteq N\setminus\{1\},|C|=\tau-1}\bigwedge_{i\in C}(\bigvee_{j=1}^{m_2} A_j^i)\Big)\triangledown\cdots\\
& \triangledown\Big(\bigvee_{C\subseteq N\setminus\{1,2\},|C|=\tau-2}\bigwedge_{i\in C}(\bigvee_{j=1}^{m_3} A_j^i)\Big)\triangledown\cdots\triangledown\Big(\bigvee_{C\subseteq N\setminus\{1,\cdots,\tau-1\},|C|=1}\bigwedge_{i\in C}(\bigvee_{j=1}^{m_\tau} A_j^i)\Big)
\end{aligned}
$$

The rule proceeds in a similar way of F_{max}: it checks if there is any subset of agents $C \subseteq N$ such that (i) the size of this subset is equal to τ, and (ii) the most important reasons of every agent belonging to C is satisfied. All possible subsets are evaluated (the first disjunction appearing in the first line). If no subset has been satisfied then we repeat the process except that we consider their first two most important reasons (the second disjunction appearing in the first line). The process continues until some alternative satisfies τ agents (lines 2 and 3). The last two lines consider the case that individual formulas may have different lengths. This rule is a generalization of the majority rule as well as the other two rules. Specifically,

- The simple majority rule, denoted by F_{maj}, can be encoded by setting quota $\tau = \lceil \frac{n+1}{2} \rceil$[3].
- The grounded rule can be encoded by setting $\tau = 1$.
- The maximal rule can be encoded by setting $\tau = n$.

To illustrate how these rules work, let us go back to Example 6.2.

Example 6.2 (*continued.*) *The model of this example is given as follows:* $\Phi_0 = \{Fiction, Comedy, Cartoon\}$ *and* $N = \{Ann,\ Kate, Bill\}$. *The set of feasible alternatives* $W = \{Gravity,\ Flipped,\ Frozen\}$*, where*

[3] $\lceil \rho \rceil$ is defined as the smallest integer greater or equal to number ρ.

$$\begin{aligned} Gravity &= \{Fiction, \neg Comedy, \neg Cartoon\}; \\ Flipped &= \{\neg Fiction, \neg Cartoon, Comedy\}; \\ Frozen &= \{\neg Fiction, Comedy, Cartoon\}. \end{aligned}$$

The prioritized reasons of each agent are described as follows:

$$\begin{aligned} \varphi_{Ann} &= (Comedy \wedge Cartoon) \triangledown Comedy \triangledown Fiction \\ \varphi_{Kate} &= (Cartoon \wedge \neg Fiction) \triangledown Comedy \triangledown Fiction \\ \varphi_{Bill} &= Fiction \triangledown \neg Cartoon \triangledown \top \end{aligned}$$

The collective formulas generated by the grounded rule, the maximal rule and the simple majority rule are respectively calculated as follows: let Φ *denote the profile* $\langle \varphi_{Ann}, \varphi_{Bill}, \varphi_{Jim} \rangle$*, then*

- $F_{grd}(\Phi) = (Fiction \vee Cartoon) \triangledown (Comedy \vee \neg Cartoon) \triangledown Fiction.$
- $F_{max}(\Phi) = \bot \triangledown (Comedy \wedge (Fiction \vee \neg Cartoon)) \triangledown (Comedy \vee Fiction \vee Cartoon).$
- $F_{maj}(\Phi) = (Comedy \wedge Cartoon) \triangledown Comedy \triangledown (Comedy \vee Fiction \vee Cartoon).$

It follows that the three aggregate formulas determine the following choice sets.

- $C(W, F_{grd}(\Phi)) = \{Gravity, Frozen\};$
- $C(W, F_{max}(\Phi)) = \{Flipped\};$
- $C(W, F_{maj}(\Phi)) = \{Frozen\}.$

That is, according to the grounded rule, the three friends are expected to choose either Gravity or Frozen; according to the maximal rule, Flipped is the collective choice, and according to the simple majority rule, they should choose Frozen.

As stressed by the previous Example, the behaviors of these rules are different. Thus, deciding which rule should be used for collective decision-making is normally situation-dependent.

Let us now provide the main result: uniform quota rules satisfy Universal domain, Anonymity, Monotonicity and Pareto principle.

Theorem 6.8. *The uniform quota rule F_τ satisfies* ***UAMP****.*

Proof. It is straightforward by the definition that F_τ satisfies Collective Completeness and **U**. We next show F_τ satisfies **A**, **M** and **P**.

(**A**). For any profile $\Phi = \langle \varphi_i \rangle_{i \in N}$ and any permutation $\sigma : N \to N$, let $F_\tau(\langle \varphi_i \rangle_{i \in N}) = A_1 \triangledown \cdots \triangledown A_m$ and $F_\tau(\langle \varphi_{\sigma(i)} \rangle_{i \in N}) = B_1 \triangledown \cdots \triangledown B_n$. By Proposition 6.6, it suffices to show that

1. $m = n$,
2. $W \models A_1 \leftrightarrow B_1$, and
3. $W \models (\bigwedge_{i=1}^{k-1} \neg A_i \wedge A_k) \leftrightarrow (\bigwedge_{i=1}^{k-1} \neg B_i \wedge B_k)$ for any $2 \leq k \leq m$.

We first show $m = n$. Suppose not for a contradiction that $m \neq n$, then either $m < n$ or $n < m$. If $m < n$, without loss of generalization, let $n = m+1$, then according to the definition of F_τ, there is some $a \in N$ such that $\varphi_a = B_1^a \triangledown \cdots \triangledown B_{m+1}^a$, but for any $\varphi_i = A_1^i \triangledown \cdots \triangledown A_{m^i}^i$, we have $m^i < m + 1$, contradicting with the assumption that σ is a permutation of N. The other case is proved in a similar way.

We next show $W \models A_1 \leftrightarrow B_1$. Suppose not for a contradiction that $W \not\models A_1 \leftrightarrow B_1$, then there is some w such that $W, w \not\models A_1 \leftrightarrow B_1$, then either ($W, w \models A_1$ and $W, w \not\models B_1$) or ($W, w \not\models A_1$ and $W, w \models B_1$). If $W, w \models A_1$ and $W, w \not\models B_1$, then by the definition of F_τ, we get $W, w \models \bigvee_{C \subseteq N, |C| = \tau} \bigwedge_{i \in C} A_1^i$ and $W, w \not\models$

$\bigvee_{C\subseteq N,|C|=\tau}\bigwedge_{i\in C} A_1^{\sigma(i)}$, contradicting with the assumption that σ is a permutation of N. The other case is proved in a similar way.

We finally show that Condition 3 holds by induction on k.

- For $k = 2$, suppose for a contradiction that $W \not\models \neg A_1 \wedge A_2 \leftrightarrow \neg B_1 \wedge B_2$, then there is some $w \in W$ such that $W, w \not\models \neg A_1 \wedge A_2 \leftrightarrow \neg B_1 \wedge B_2$. Say $W, w \models \neg A_1 \wedge A_2$ and $W, w \not\models \neg B_1 \wedge B_2$. Then by Condition 2, we have $W, w \models \neg B_1$, then $W, w \models \neg B_2$ and $W, w \models A_2$. But by the definition of F_τ, we have that $A_2 = \bigvee_{C\subseteq N,|C|=\tau}\bigwedge_{i\in C}(A_1^i \vee A_2^i)$ and $B_2 = \bigvee_{C\subseteq N,|C|=\tau}\bigwedge_{i\in C}(A_1^{\sigma(i)} \vee A_2^{\sigma(i)})$. Since σ is a permutation of N, so $W, w \models A_2 \leftrightarrow B_2$: a contradiction. The other case is proved in a similar way.

- For $k = l + 1$, we need the following claim.

  Claim 6.9. Given Condition 2 and for any $2 \leq l \leq m$, if $W \models \bigwedge_{t=2}^{l}\big((\bigwedge_{i=1}^{t-1} \neg A_i \wedge A_t) \leftrightarrow (\bigwedge_{i=1}^{t-1} \neg B_i \wedge B_t)\big)$, then $W \models \bigwedge_{i=1}^{l} \neg A_i \leftrightarrow \bigwedge_{i=1}^{l} \neg B_i$.

  We prove this claim by induction on l.

  For$l = 2$, suppose that $W \models A_1 \leftrightarrow B_1$ and $W \models (\neg A_1 \wedge A_2) \leftrightarrow (\neg B_1 \wedge B_2)$, we need to show $W \models (\neg A_1 \wedge \neg A_2) \leftrightarrow (\neg B_1 \wedge \neg B_2)$. Further assume that $W, w \models \neg A_1 \wedge \neg A_2$ for any $w \in W$, then $W, w \models \neg A_1$ and $W, w \models \neg A_2$. And by assumption, we get $W, w \models \neg B_1$ and $W, w \not\models \neg B_1 \wedge B_2$, so $W, w \models \neg B_2$. The other direction is proved in a similar way.

  For $l = r + 1$, assume that $W \models A_1 \leftrightarrow B_1$ and $W \models \bigwedge_{t=2}^{r+1}\big((\bigwedge_{i=1}^{t-1} \neg A_i \wedge A_t) \leftrightarrow (\bigwedge_{i=1}^{t-1} \neg B_i \wedge B_t)\big)$, we need to show $W \models \bigwedge_{i=1}^{r+1} \neg A_i \leftrightarrow \bigwedge_{i=1}^{r+1} \neg B_i$. Further suppose $W, w \models \bigwedge_{i=1}^{r+1} \neg A_i$, then $W, w \models \bigwedge_{i=1}^{r} \neg A_i$. And by induction hypothesis, we get $W, w \models \bigwedge_{i=1}^{r} \neg B_i$. Since σ is a permutation of N, so by the definition of F, $W \models A_{r+1} \leftrightarrow B_{r+1}$. And by assumption $W, w \models \neg A_{r+1}$ we get $W, w \models \neg B_{r+1}$. Thus, $W, w \models \bigwedge_{i=1}^{r+1} \neg B_i$. The other direction is proved in a similar way.

With this claim, we now prove the inductive case as follows: Suppose not for a contradiction that $W \not\models (\bigwedge_{i=1}^{l} \neg A_i \wedge A_{l+1}) \leftrightarrow (\bigwedge_{i=1}^{l} \neg B_i \wedge B_{l+1})$, then there is $w \in W$ such that $W, w \not\models (\bigwedge_{i=1}^{l} \neg A_i \wedge A_{l+1}) \leftrightarrow (\bigwedge_{i=1}^{l} \neg B_i \wedge B_{l+1})$, so either $(W, w \models \bigwedge_{i=1}^{l} \neg A_i \wedge A_{l+1}$ and $W, w \not\models \bigwedge_{i=1}^{l} \neg B_i \wedge B_{l+1})$ or $(W, w \not\models \bigwedge_{i=1}^{l} \neg A_i \wedge A_{l+1}$ and $W, w \models \bigwedge_{i=1}^{l} \neg B_i \wedge B_{l+1})$. If $W, w \models \bigwedge_{i=1}^{l} \neg A_i \wedge A_{l+1}$ and $W, w \not\models \bigwedge_{i=1}^{l} \neg B_i \wedge B_{l+1}$, then $W, w \models \bigwedge_{i=1}^{l} \neg A_i$. And by induction hypothesis and Claim 6.9, we get $W, w \models \bigwedge_{i=1}^{l} \neg B_i$. Then by assumption we obtain that $W, w \models A_{l+1}$ and $W, w \not\models B_{l+1}$. Without loss of generalization, consider $m^i \geq l+1$ and $n^i \geq l+1$ for all $i \in N$. Then according to the definition of F_τ, $W, w \models \bigvee_{C \subseteq N, |C| = \tau} \bigwedge_{i \in C} \bigvee_{j=1}^{l+1} A_j^i$ and $W, w \not\models \bigvee_{C \subseteq N, |C| = \tau} \bigwedge_{i \in C} \bigvee_{j=1}^{l+1} A_j^{\sigma(i)}$, contradicting with the assumption that σ is a permutation of N. The other case is proved in a similar way.

Thus, $\Vdash_W F_\tau(\langle \varphi_i \rangle_{i \in N}) \equiv F_\tau(\langle \varphi_{\sigma(i)} \rangle_{i \in N})$.

(**M**). For any two profiles $\Phi = \langle \varphi_i \rangle_{i \in N}$, $\Phi' = \langle \varphi'_i \rangle_{i \in N}$ and for all $i \in N$, assume $\varphi_i \Vdash_W \varphi'_i$, we need to show $F_\tau(\Phi) \Vdash_W F_\tau(\Phi')$. Let $F_\tau(\langle \varphi_i \rangle_{i \in N}) = A_1 \triangledown \cdots \triangledown A_m$ and $F_\tau(\langle \varphi' \rangle_{i \in N}) = B_1 \triangledown \cdots \triangledown B_r$. By Definition 6.7, it suffices to show

1. $m \geq r$, and
2. $W \models A_k \to \bigvee_{i=1}^{k} B_i$ for any $1 \leq k \leq r$.

Let $\varphi_i = A_1^i \triangledown \cdots \triangledown A_{m^i}^i$ and $\varphi'_i = B_1^i \triangledown \cdots \triangledown B_{r^i}^i$ for any $i \in N$. By assumption we get $m^i \geq r^i$ for any $i \in N$, then by the definition of F_τ, we have $m \geq r$.

We next show the second condition by induction on k.

Consider the base case when $k = 1$. By the assumption we get $W \models A_1^i \to B_1^i$ for any $i \in N$, then $W \models \bigvee_{C \subseteq N, |C| = \tau} \bigwedge_{i \in C} A_1^i \to \bigvee_{C \subseteq N, |C| = \tau} \bigwedge_{i \in C} B_1^i$, i.e., $W \models A_1 \to B_1$.

For the induction step $t+1$, suppose not for a contradiction that $W \not\models A_{t+1} \to$

$\bigvee_{j=1}^{t+1} B_j$, then there is some $w \in W$ such that $W, w \not\models A_{t+1} \to \bigvee_{j=1}^{t+1} B_j$, then $W, w \models A_{t+1}$ and $W, w \not\models \bigvee_{j=1}^{t+1} B_j$, so $W, w \not\models B_{t+1}$. By the definition of F_τ and from $W, w \models A_{t+1}$, we have $W, w \models \bigvee_{C \subseteq N, |C|=\tau} \bigwedge_{i \in C} \bigvee_{j=1}^{t+1} A^i_j$, then there is some $C \subseteq N$ with $|C| = \tau$ such that $W, w \models \bigvee_{j=1}^{t+1} A^i_j$ for any $i \in C$, so for any $i \in C$ there is $1 \leq d_i \leq t+1$ such that $W, w \models A^i_{d_i}$. And by the assumption we get for any $i \in C$ there is $1 \leq d_i \leq t+1$ such that $W, w \models \bigvee_{j=1}^{d_i} B^i_j$, so $W, w \models \bigwedge_{i \in C, |C|=\tau} \bigvee_{j=1}^{t+1} B^i_j$. Thus, by the definition of F_τ, $W, w \models B_{t+1}$, contradicting with $W, w \not\models B_{t+1}$.

(**P**). For any profile $\Phi = \langle \varphi_i \rangle_{i \in N}$ and for any complete $\varphi \in \mathcal{L}_{RCL}$, assume $\models_W \varphi_i \equiv \varphi$ for any $i \in N$. Let $F_\tau(\langle \varphi_i \rangle_{i \in N}) = A_1 \triangledown \cdots \triangledown A_m$ and $\varphi = B_1 \triangledown \cdots \triangledown B_r$. By Proposition 6.6, it suffices to show that

1. $m = r$,
2. $W \models A_1 \leftrightarrow B_1$, and
3. $W \models (\bigwedge_{i=1}^{k-1} \neg A_i \wedge A_k) \leftrightarrow (\bigwedge_{i=1}^{k-1} \neg B_i \wedge B_k)$ for any $2 \leq k \leq m$.

Since $m^i = r$ any $i \in N$, so $m = r$ by the definition of F.

By assumption that $\models_W \varphi_i \equiv \varphi$ for any $i \in N$, we obtain that $W \models A^i_1 \leftrightarrow B_1$, then $W \models \bigvee_{C \subseteq N, |C|=\tau} \bigwedge_{i \in C} A^i_1 \leftrightarrow B_1$, i.e., $W \models A_1 \leftrightarrow B_1$.

We show the third condition by induction on k.

For $k = 2$, assume $W, w \models \neg A_1 \wedge A_2$, then by Condition 2, we get $W, w \models \neg B_1$, so by assumption $W, w \models \bigwedge_{i=1}^{n} \neg A^i_1$. And by the definition of F_τ, we get $W, w \models \bigvee_{C \subseteq N, |C|=\tau} \bigwedge_{i \in C} (A^i_1 \vee A^i_2)$, then $W, w \models \bigvee_{C \subseteq N, |C|=\tau} \bigwedge_{i \in C} A^i_2$. And by the assumption, so $W, w \models \neg B_1 \wedge B_2$. The other direction is proved in a similar way.

For $l+1$, suppose not for a contradiction that $W \not\models (\bigwedge_{i=1}^{l} \neg A_i \wedge A_{l+1}) \leftrightarrow (\bigwedge_{i=1}^{l} \neg B_i \wedge B_{l+1})$, then there is $w \in W$ such that $W, w \not\models (\bigwedge_{i=1}^{l} \neg A_i \wedge A_{l+1}) \leftrightarrow (\bigwedge_{i=1}^{l} \neg B_i \wedge B_{l+1})$. Say $W, w \models \bigwedge_{i=1}^{l} \neg A_i \wedge A_{l+1}$ and $W, w \not\models \bigwedge_{i=1}^{l} \neg B_i \wedge B_{l+1}$. Then $W, w \models \bigwedge_{i=1}^{l} \neg A_i$. And by induction hypothesis and Claim 6.9, we get $W, w \models \bigwedge_{i=1}^{l} \neg B_i$, then by assumption we obtain that $W, w \models A_{l+1}$ and $W, w \not\models B_{l+1}$. And by the assumption that $\models_W \varphi_a \equiv \varphi$ for any $a \in N$, we get $W, w \not\models \bigwedge_{i=1}^{l+1} \neg A_i^a$, so by the definition of F_τ, we get $W, w \not\models A_{l+1}$: a contradiction. The other direction is proved in a similar way.

Therefore, the uniform quota rule F_τ satisfies **UAMP**. □

Corollary 6.10. *The grounded rule F_{grd}, the maximal rule F_{max} and the simple majority rule F_{maj} satisfy **UAMP**.*

It should be noted that the size of an aggregate formula can be significantly reduced via the following two ways. Firstly, any aggregate formula represents a preference ordering over alternative, and thus its size is virtually only determined by the number of alternatives as well as their properties, and has nothing to do with the number of individuals. As shown in Example 6.2, prioritized disjuncts can be extremely simplified by equivalence laws of propositional logic. Secondly, there are many reasons that none of the alternatives in W satisfies, and if we remove such "dummy" reasons, the collective preference ordering over alternatives will not be changed, as they correspond to the empty choice set. For instance, in Example 6.2, by removing the "dummy" reasons, the aggregate formula $F_{max}(\langle \varphi_{Ann}, \varphi_{Bill}, \varphi_{Jim} \rangle)$ is simplified as $(Comedy \wedge (Fiction \vee \neg Cartoon)) \triangledown (Comedy \vee Fiction \vee Cartoon)$, which represents the same preference ordering, i.e., *Gravity* and *Frozen* are indifferent, and *Flipped* is better than any of them. Unfortunately, the size of an aggregate formula would be exponential in the worst case.

6.3.3 Model Checking

One of the advantages of RCL is that not only individual preferences but also the collective choice rules are built into the logic, which allows us to use the model-checking techniques to automatically generate individual and collective choices. The *model-checking problem* for RCL is the following problem: for a given RCL formula φ, a set W of alternatives and an alternative $w \in W$, determine whether $W, w \models \varphi$ or not.

Proposition 6.11. *There is an algorithm that runs in time $O(|W| \times \|\varphi\|)$ to check, given any set $W \subseteq 2^{\Phi_0}$, any alternative $w \in W$ and any formula $\varphi \in \mathcal{L}_{RCL}$, whether $W, w \models \varphi$, where $\|\varphi\|$ denotes the size of φ, i.e., the number of symbols occurring in φ.*

Proof. It suffices to develop an algorithm that runs in time $O(|W| \times \|\varphi\|)$ to compute the set of alternatives $w' \in W$ such that $W, w' \models \varphi$. We implement this computation by Algorithm 6.3.1. The general idea is to compute the desired choice sets for all subformulas of φ recursively. Given any subformula ψ of φ, assuming that the mentioned choice sets for all proper subformulas of ψ are available, the computation on ψ can be done in time $O(|W|)$. The number of subformulas of φ is clearly not greater than $\|\varphi\|$. Thus, Algorithm 6.3.1 must terminate in time $O(|W| \times \|\varphi\|)$. According to the semantics, it is also easy to verify the correctness of this algorithm. $\square$

6.4 Discussion and Summary

In this section, we first embed RCL into the standard modal logic S5 [Blackburn et al., 2002], and investigate the relation between collective choice and preference aggregation, then discuss some related work, and finally conclude this chapter with some future work.

6.4.1 Embeddings of RCL to S5

On the semantical level, the model of RCL can be easily regarded as an universal Kripke model. Given a set W of alternatives, we construct an universal Kripke

Algorithm 6.3.1: computeTruth(W, φ)

```
Input  : W ⊆ 2^{Φ_0} and φ ∈ L_RCL
Output: the set C(W, φ)
begin
    switch φ do
        case p, where p ∈ Φ_0
            T ← {w ∈ W : p ∈ w};
            break;
        case ¬ψ, where ψ ∈ L_RCL
            T ← computeTruth(W, ψ);
            T ← W \ T;
            break;
        case ψ_1 ∧ ψ_2, where ψ_1, ψ_2 ∈ L_RCL
            T ← computeTruth(W, ψ_1);
            T ← T ∩ computeTruth(W, ψ_2);
            break;
        case ψ_1 ▽ ψ_2, where ψ_1, ψ_2 ∈ L_RCL
            T ← computeTruth(W, ψ_1);
            if T = ∅ then
                T ← computeTruth(W, ψ_2);
            end
            break;
        otherwise
            T ← ∅;
        endsw
    endsw
    return T;
end
```

model $M^W = (S, R, v)$ as follows: let $W = S$, $R = S \times S$ and $v(p) = \{s \in S \mid p \in s\}$. On the syntactic level, the prioritized connective can be defined by the global modality A [Blackburn et al., 2002]. It is interpreted as follows: Given an universal Kripke model $M = (S, R, v)$ and a state $s \in S$,

$$M, s \models A\varphi \text{ iff } \quad \text{for all } t \in S, M, t \models \varphi$$

We next define a translation map from RCL formulas to S5 formulas to make RCL embedded into S5.

Definition 6.12. A translation Tr from RCL formulas to S5 formulas is defined as follows:

- $\mathrm{Tr}(A) = A \quad$ for all propositional formulas A
- $\mathrm{Tr}(\varphi_1 \nabla \varphi_2) = \mathrm{Tr}(\varphi_1) \vee (A\neg\mathrm{Tr}(\varphi_1) \wedge \mathrm{Tr}(\varphi_2))$

We have the following correspondent result with respect to the translation.

Proposition 6.12. *Given a set of alternatives W and $w \in W$, for any RCL formula $\varphi \in \mathcal{L}$,*

$$W, w \models \varphi \text{ iff } M^W, w \models Tr(\varphi)$$

Proof. It is proved by induction on the structure of φ. It is trivial for the case when φ is a propositional formula. Here we just show the case for the prioritized formula.

$$\begin{aligned}
&\text{Assume } W, w \models (\varphi_1 \nabla \varphi_2) \\
&\text{iff} \\
&W, w \models \varphi_1, \text{ or (for all } v \in W, W, v \not\models \varphi_1 \text{ and } W, w \models \varphi_2) \\
&\text{iff} \\
&M^W, w \models \mathrm{Tr}(\varphi_1), \text{ or (for all } v \in R(w), M^W, v \not\models \mathrm{Tr}(\varphi_1) \\
&\text{and } M^W, w \models \mathrm{Tr}(\varphi_2)) \\
&\text{iff} \\
&M^W, w \models \mathrm{Tr}(\varphi_1) \vee (A\neg\mathrm{Tr}(\varphi_1) \wedge \mathrm{Tr}(\varphi_2)) \\
&\text{iff} \\
&M^W, w \models \mathrm{Tr}(\varphi_1 \nabla \varphi_2)
\end{aligned}$$

□

6.4.2 Relation with Preference Aggregation

Let us begin by showing that the Condorcet's paradox [Gehrlein, 1983] can be naturally avoided in RCL due to the characteristics of the language. As it is shown in Proposition 6.2, any formula generates a preference ordering over the alternatives on the basis of the priorities over reasons. This guarantees that all collective rules in RCL always induce a preference ordering over alternatives. For example, given the set of atomic properties $\Phi_0 = \{p, q, r\}$, consider three candidates $W = \{w_1, w_2, w_3\}$, where $w_1 = \{p, \neg q, \neg r\}$, $w_2 = \{\neg p, q, \neg r\}$ and $w_3 = \{\neg p, \neg q, r\}$. There are three voters 1, 2 and 3 with their voting reasons given as follows: $\varphi_1 = p \triangledown q \triangledown r$, $\varphi_2 = q \triangledown r \triangledown p$, and $\varphi_3 = r \triangledown p \triangledown q$. According to the simple majority rule ($\tau = 2$), we get the collective result as follows:

$$F_{maj}(\langle \varphi_1, \varphi_2, \varphi_3 \rangle) = [(p \wedge q) \vee (p \wedge r) \vee (q \wedge r)] \triangledown [((p \vee q) \wedge (q \vee r)) \vee ((p \vee q) \wedge (p \vee r)) \vee ((q \vee r) \wedge (p \vee r))] \triangledown (p \vee q \vee r)$$

Instead of generating a cyclic (intransitive) preference ordering, this aggregate formula would result in a tie with three alternatives indifferent, i.e., $C(W, F_2(\langle \varphi_1, \varphi_2, \varphi_3 \rangle)) = \{w_1, w_2, w_3\}$.

As mentioned, all Arrowian conditions are plausible under the new setting except Independence of irrelevant alternatives (IIA). The main reason is that different from other Arrowian conditions, IIA is a specification about the order of two particular alternatives. This condition requires for any two alternatives w and w', the collective preference between w and w' depends only upon individual preferences over that pair. On the other hand, a RCL-formula always specifies a priority relation over reasons which further induces a preference ordering over all alternatives. Thus, it is impossible for a RCL- formula to describe two particular alternatives, which provides a natural way to circumvents Arrow's impossibility result in RCL.

In addition, we consider another condition Monotonicity which is the qualitative counterpart of the monotonicity in database aggregation algorithm [Fagin et al., 2003a]. This condition is an important property for rank aggregation rules. In fact, the proposed rules may be regarded as rank aggregation rules, since they aggregate reasons according to their priority layer by layer. It turns out that all other Arrowian conditions are consistent with Monotonicity.

6.4.3 Related Work

In recent years, many logical frameworks have been proposed for representing and reasoning about choices and preferences [Boutilier, 1994, Brewka et al., 2004, Lafage and Lang, 2000, Liu, 2008, Osherson and Weinstein, 2012, Pedersen et al., 2013, Tan and Pearl, 1994, van Benthem et al., 2009, van Ditmarsch et al., 2007]. Most of the previous work takes preferences as fundamental and primitive concepts, and typically treats them as modal operators [van Benthem et al., 2009, van Ditmarsch et al., 2007]. On the one hand, these preference logics mainly focus on investigating logical properties of preferences with little concern about how preferences are formed. On the other hand, formulas in these logics are interpreted by an arbitrary given (utilitarian or ordinal) preference relation, which does not provide a facility to represent different preference orderings.

Recent work in [Brewka et al., 2004, Liu, 2008, Osherson and Weinstein, 2012, Pedersen et al., 2013, van Benthem et al., 2014] takes a different angle by exploring how preferences come from.

Liu [2008] introduces *a priority base* that is ordered by the importance and also discusses the ways to rationally derive preferences from it. Unfortunately, the priority base is defined in the semantic level and the priority operator is not used for representing individual and collective choices. Recently, van Benthem et al. [2014] investigate in details how priority bases and preference logics are related in a deontic context.

Pedersen et al. [2013] develop a modal logic for reasoning about the framework of Dietrich and List [2013b], and show how to use a standard modal logic for reasoning about reason-based preferences. Different from RCL, they use the standard modal language and only encode the priority (weighing relation) over reasons into the semantic model. In particular, they generalize their work to the multi-agent case mainly for modelling non-trivial properties such as disagreement, consensus in a multi-agent setting rather than for aggregating preferences. Moreover, they assume that all agents share the same priority over reasons. This means all agents have the same preference ordering over alternatives, which is unsuitable for preference aggregation.

Osherson and Weinstein [2012] propose another logical formalism for reasoning about reason-based preferences. Different from our qualitative approach, their logic is a non-standard modal logic built in the context of utility, and each alternative can be evaluated according to various utility scales. Moreover, they mainly consider different ways to combine utilities induced by different reasons for single agent without generalizing their work to the multi-agent dimension.

One of the closest related work is probably the framework of Brewka et al. [2004]. The non-standard part of this logic is a logical connective $\vec{\times}$ called *ordered disjunction*. The intuition behind $\triangledown$ is similar to that of the ordered disjunction. Also, the same idea, though applied to deontic reasoning, is independently developed in [Governatori and Rotolo, 2006]. Different from our motivation and approaches, they propose a nonmonotonic formalism for representing reason-based choices and do not consider the multi-agent case.

Based on above analysis, we may find that few logical formalisms can not only represent reason-based preferences, but also provide efficient decision procedures so as to automatically generate individual and collective choices based on reasons. To the best of our knowledge, [Lafage and Lang, 2000] is the only one that can do both. It uses weighted logics for representing preferences, that is, each agent expresses her preferences by means of logical formulas weighted by importance degrees, and then generates the collective result by calculating the utility of each agent into a collective utility function. The major difference from this work is that we use a qualitative approach, and show that preferences as well as collective rules are expressed in a standard modal logic, which allows us to develop a model-checking algorithm to generate individual choices and collective choices.

In summary, this chapter has proposed a modal logic for representing and reasoning about individual and collective choices based on reasons. Not only individual preferences but also collective choice rules are expressed within this logic. We have then developed a model checking algorithm to generate individual and collective choices. Based on the proposed collective rules, we have demonstrated that all Arrowian conditions are plausible in this logic except Independence of Irrelevant Alternatives, which allows us to avoid the impossibility result.

As RCL can encode preferences and collective choice rules at the same time, our long term goal is to enable an intelligent agent to reason about collective choices. To this end, several extensions may be considered.

Firstly, to keep our logical formalism as simple and intuitive as possible, we currently do not allow the interplay of the prioritized connective and the other classical connectives. Consequently, the context dependent preferences are beyond the expressivity power of this language. It would be interesting to remove some syntactical limitations of the language for being in position to represent more general preferences [Lang, 2006].

Secondly, we want to investigate the representation results for the proposed collective choice rules. Though it is a difficult problem in social choice theory to provide characterization results for specific aggregation rules, based on some existing results [Andréka et al., 2002, Dietrich and List, 2007b, May, 1952], this is not impossible.

Finally, we have assumed that one agent's choice is given without any influence from others. However, in many situations, beliefs about other agents' choices might affect the agent's own decision, and consequently change the collective choice. Thus, it is worth extending the language with epistemic operators so as to study the effect of beliefs on individual and collective choices.

In particular, in many real-world situations, a group making collective decisions may assign individual members or subgroups different priorities to determine the collective decision. In the next chapter, we will investigate how to make collective decision under such a hierarchical environment.

Chapter 7

Judgment Aggregation Under Voters' Hierarchy

Chapter 6 proposes a modal logic for modelling reason-based collective choices in which each individual is treated equally. However, in many real-world situations, a group making collective decisions may assign individual members or subgroups different priorities to determine the collective decision. For instance, legislatures or expert panels may assign specialist members such priority so as to rely on their expertise. In this chapter, we focus on *judgment aggregation*, a relatively new topic in social choice theory, and provide logical methods to investigate how the judgment from each individual affects group judgment in such a hierarchical environment.

7.1 Background

Judgment aggregation is an interdisciplinary research topic in economics, philosophy, political science, law and recently in computer science [Brandt et al., 2012, Dietrich and List, 2007a, Konieczny and Pérez, 2002, Mongin, 1995, Pigozzi, 2006, Wilson, 1975]. It deals with the problem of *how a set of group judgments on certain issues, represented by logical propositions, can be formed based on individuals' judgments on the same issues.* Although most of voting rules for social choice, such as majority rules, unanimity rules or dictatorships, are applicable to judgment aggregation, their behaviour can be significantly different due to possible logical links among the propositions on which a collective decision has to be made. A well-known example is the so-called *doctrinal paradox* [Kornhauser and Sage, 1993, List, 2012], which shows that the majority rule fails to guarantee consistent group judgments.

Suppose a court consisting of three judges has to reach a verdict in a breach-of-contract case. There are three propositions on which the court is required to make judgments:

p: *The defendant was contractually obliged not to do a particular action.*

q: *The defendant did that action.*

r: *The defendant is liable for a breach of contract.*

According to the legal doctrine, propositions p and q are jointly necessary and sufficient for proposition r, that is $p \wedge q \leftrightarrow r$. Now the three judgments on the propositions are shown in Table 7.1. If the three judges take a majority vote on

	p	q	r
Judge 1	Yes	Yes	Yes
Judge 2	Yes	No	No
Judge 3	No	Yes	No
Majority	Yes	Yes	No

Table 7.1: A doctrinal paradox.

proposition r, the outcome is its rejection: a 'not liable' verdict. But if they take majority votes on each of p and q instead, then p and q are accepted, so by the legal doctrine, r should be accepted as well: a 'liable' verdict. Although each judge holds consistent individual judgments on the three propositions, there are majorities for p, q and $\neg r$, a logically inconsistent set of propositions with respect to the constraint $p \wedge q \leftrightarrow r$.

More significantly, List and Pettit [2002] show an impossibility result, similar to Arrow's impossibility theorem [Arrow, 1950], that no aggregation rule can generate consistent collective judgments if we require an aggregation rule to satisfy a set of "plausible" conditions. However, such an impossibility result does not discourage the investigation of judgement aggregation. None of the conditions on either aggregation rules or decision problems, is indefectible. By weakening or varying these conditions, a growing body of literature on judgement aggregation has emerged in recent years. For an overview of the related research, please refer to [List, 2012, List and Puppe, 2009].

Among all the plausible conditions that lead to impossibility results on judgment aggregation, *completeness* as one of the rationality requirements has received criticism of being overly demanding in many real-world situations, where an individual may abstain from voting on a decision issue, and a group judgment on some issue may be undetermined. In fact, if we give up completeness, we are able to circumvent impossibility [Dietrich and List, 2007c, 2008a, Dokow and Holzman, 2010, Gärdenfors, 2006, List and Pettit, 2002]. Among them, Gärdenfors [2006] proves a representation theorem for judgment aggregation without completeness, which shows that under certain fairly natural conditions, the only possible aggregation rules are oligarchic. Dietrich and List [2008a] strengthen Gärdenfors' results and show that by giving up completeness in favor of deductive closure, oligarchies instead of dictatorships are obtained.

Moreover, in many real-world decision-making settings, when a group of agents forms collective judgments, some group members or subgroups may have priority to decide certain propositions. For instance, legislatures or expert panels may assign specialist members such priority so as to rely on their expertise; when some propositions concern group members' private spheres, they may be also assigned the rights to be decisive on those propositions [Dietrich and List, 2008b]. However, in the current literature on judgment aggregation such priority has been rarely investigated. In particular, Dietrich and List [2008b] propose a generalization of Sen's 'liberal paradox' [Sen, 1970]. Under plausible conditions, the assignment of rights to two or more individuals or subgroups is inconsistent with the unanimity principle requiring unanimously accepted propositions be collectively accepted. Following this work, most existing literature is mainly concerned with liberal (im)possibility results [Dietrich and List, 2008b, Patty and Penn, 2014, van Hees, 1999]. There are seldom specific aggregation rules to formally display how to generate consistent collective judgments in such a hierarchical group.

The aim of this chapter is to investigate how individual judgements affect group judgement in a hierarchical environment. To this end, we first provide a logic-based model for judgment aggregation with abstentions, and then develop a lexicographic aggregation rule for generating consistent collective judgments based on voters' hierarchy. We also show this rule satisfies a set of plausible conditions and has a tractable computational complexity. We finally investigate the oligarchic property of this rule.

7.2 Judgment Aggregation with Abstentions

In this section, we first provide a logic-based model for judgment aggregation with abstentions, and then explore a set of plausible conditions for aggregations in terms of abstentions.

7.2.1 The Logic-Based Model

We consider a finite set of individuals $N = \{1, 2, \dots, n\}$ with $|N| \geq 2$. They face a decision problem that requires collective judgments on propositions. Propositions are represented by a logical language $\mathcal{L}$ with a set Φ_0 of propositional variables and standard logical connectives $\{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$. Following [List and Pettit, 2002, 2004], we assume that the underlying logic is the classical propositional logic with standard syntax and semantics. The set of literals, denoted by $\mathcal{P}$, consists of either propositional variables or their negations, i.e., $\mathcal{P} = \{p, \neg p \mid p \in \Phi_0\}$.

The set of propositions on which judgments are to be made is called the *agenda*. Formally, the agenda is a finite non-empty subset $X \subseteq \mathcal{L}$ closed under negation (i.e., if $\varphi \in X$, then $\neg\varphi \in X$), and under propositional variables (i.e., for all $\varphi \in \mathcal{L}$, if $\varphi \in X$, then for all $p \in \Phi_0$ occur in φ, $p \in X$). Similar to [Dietrich and List, 2008a], we assume that double negations in the agenda cancel each other. That is, $X = \{\varphi, \neg\varphi : \varphi \in X^*\}$, where $X^* \subseteq \mathcal{L}$ is a set of unnegated propositions. Let $X_0 = X \cap \mathcal{P}$ be the set of literals included in the agenda. Consider the doctrinal paradox in Section 1. The agenda is

$$\{p, q, p \wedge q, \neg p, \neg q, \neg(p \wedge q)\}.$$

The set of literals in the agenda is

$$\{p, q, \neg p, \neg q\}.$$

Note that for the sake of readability, we replace r by $p \wedge q$ as they are logically equivalent. We call a set $Y \subseteq \mathcal{L}$ *minimally inconsistent* if it is inconsistent and every proper subset of Y is consistent. For instance, $\{p, q, \neg(p \wedge q)\}$ is a minimally inconsistent set, while $\{p, \neg q, p \wedge q\}$ is not, since its proper subset $\{\neg q, p \wedge q\}$ is inconsistent. The agenda X is *non-simple* if it has a minimal inconsistent subset Y

such that $|Y| \geq 3$. For instance, the agenda in the doctrinal paradox is non-simple, since it includes a minimal inconsistent subset $\{p, q, \neg(p \wedge q)\}$.

We represent each *individual judgment set* as a subset of the agenda, which indicates all the propositions that this individual accepts or believes to be true. Formally, individual i's judgment set, denoted by Φ_i, is a subset of X, i.e., $\Phi_i \subseteq X$. We assume that each individual judgment set Φ_i satisfies the following conditions:

- *Consistence*: all its members can be simultaneously true, i.e., $\Phi_i \not\models \perp$.
- *Deductive closure*: for every $\varphi \in X$, if $\Phi_i \models \varphi$, then $\varphi \in \Phi_i$.

As we have mentioned in Introduction, we do not require Φ_i to be complete, i.e., for every pair $\varphi, \neg\varphi \in X$, either $\varphi \in \Phi_i$ or $\neg\varphi \in \Phi_i$. Then for each proposition $\varphi \in X$, it may happen that $\varphi \notin \Phi_i$ and $\neg\varphi \notin \Phi_i$. In this case, we say that individual i abstains from making a judgment on φ. Given a proposition $\varphi \in X$, individual i's judgment on this proposition may be acceptance (i.e.,$\varphi \in \Phi_i$), rejection (i.e.,$\neg\varphi \in \Phi_i$) and abstention, denoted as $+$, $-$ and $\#$, respectively. With abstentions, if an individual makes her judgments over φ and ψ, respectively, then her judgments on their compound formulas must be consistent with the judgments of φ and ψ. The composition of two judgments with respect to $\to$ is depicted in Table 7.2. For

$\varphi \to \psi$		ψ		
		$+$	$\#$	$-$
φ	$+$	$+$	$\#$	$-$
	$\#$	$+$	$\#$	$\#$
	$-$	$+$	$+$	$+$

TABLE 7.2: The compositions for $\to$.

instance, if an individual abstains from voting on φ (i.e.,$\#$) and rejects ψ (i.e.,$-$), then she should abstain from voting on $\varphi \to \psi$ (i.e.,$\#$). Otherwise, her individual judgment set would be inconsistent. Given each individual judgment set Φ_i, the vector $\langle \Phi_i \rangle_{i \in N}$ is called *a profile of individual judgment sets*. For instance, in the doctrinal paradox, the profile is

$$\langle\{p, q, p \wedge q\}, \{p, \neg q, \neg(p \wedge q)\}, \{\neg p, q, \neg(p \wedge q)\}\rangle.$$

Finally, a *judgment aggregation rule* is a function F that assigns to each profile $\langle\Phi_i\rangle_{i\in N}$ a single collective judgment set $\Phi \subseteq X$, where $\varphi \in \Phi$ means that the group as a whole accepts φ. The set of all admissible profiles is called the *domain* of F, denoted by $Dom(F)$. Note that we do not require the collective judgments to be complete, which allows that a group may abstain from voting on certain propositions. Below we will impose plausible conditions on aggregation rules. A standard example of aggregation rules is *the majority rule*, as introduced in the doctrinal paradox, where each proposition is collectively accepted if and only if the number of individuals accepted this proposition exceeds the half, i.e.,

$$F(\langle\Phi_i\rangle_{i\in N}) = \{\varphi \in X \ : \ |i \in N : \varphi \in \Phi_i| > n/2\}.$$

7.2.2 Conditions on Aggregation Rules

We begin with a simple impossibility result generated by minimal conditions in this new context, and then explore plausible conditions under which the impossibility does not arise.

7.2.2.1 An Impossibility Result

The impossibility result holds for all agendas exhibiting "mild" interconnections in the following sense. We call a set $Y \subseteq \mathcal{L}$ *minimally inconsistent* if it is inconsistent and every proper subset of Y is consistent. The agenda X is *non-simple* if it has a minimal inconsistent subset Y such that $|Y| \geq 3$.

Let F be an aggregation rule. Consider the following conditions:

Universal Domain (UD). The domain of F includes all profiles of consistent individual judgment sets.

Non-dictatorship (ND). *There is no* $x \in N$ *such that for all* $\langle \Phi_i \rangle_{i \in N} \in Dom(F)$, $F(\langle \Phi_i \rangle_{i \in N}) = \Phi_x$. This is a basic democratic requirement: no single individual should always determine the collective judgment set.

Unanimity with Abstentions (U). *For every* $\varphi \in X$*, if there is some* $V \subseteq N$ *such that* $V \neq \emptyset$*, for all* $i \in V$*.* $\varphi \in \Phi_i$ *and for all* $j \in N \backslash V$*.* $\varphi \# \Phi_j$, then $\varphi \in F(\langle \Phi_i \rangle_{i \in N})$. That is, if a nonempty set of individuals agrees on accepting a proposition φ, while all the others abstain from voting on φ, then this condition requires that the group should accept φ as well.

It is worth noting that the unanimity with abstentions is the counterpart of condition *be unanimous with abstentions* in [Andréka et al., 2002] and the *Pareto optimality* in [Gärdenfors, 2006] (also called *unanimity principle* in [Dietrich and List, 2008b], *Paretian condition* in [Dokow and Holzman, 2010]).

Proposition 7.1. *If* F *satisfies unanimity with abstentions, then* F *is Pareto optimal, i.e.,* for every $\varphi \in X$, if $\varphi \in \Phi_i$ for every $i \in N$, then $\varphi \in F(\langle \Phi_i \rangle_{i \in N})$.

The next proposition says that *non-dictatorship* can be derived from unanimity with abstentions.

Proposition 7.2. *Every judgment aggregation rule satisfying unanimity with abstentions is non-dictatorial.*

Proof. Assume that F is dictatorial in some individual $x \in N$, then $N \backslash \{x\} \neq \emptyset$. Take $\varphi \in X$ and define $\varphi \# \Phi_x$ and $\varphi \in \Phi_i$ for every $i \in N \backslash \{x\}$. By unanimity with abstentions, $\varphi \in F(\langle \Phi_i \rangle_{i \in N})$, then $F(\langle \Phi_i \rangle_{i \in N}) \neq \Phi_x$, contradiction. □

On the other hand, the following example shows that this condition fails to guarantee a consistent collective judgment set.

Example 7.1. *Suppose Ann, Bill and Tom have to make group judgments on three logically connected propositions as follows:*

p: There is the elixir of life.

q: Humans can be immortal.

$p \to q$*: If there is the elixir of life, then humans can be immortal.*

Their individual judgments are shown in Table 7.3.

	p	q	p→q
Ann	+	#	#
Bill	#	−	#
Tom	#	#	+

TABLE 7.3: Individual judgments in Example 7.1.

According to the unanimity with abstentions, the collective judgment set is $\{p, p \to q, \neg q\}$ which is inconsistent. In fact, the following result shows that this is not a single case.

Theorem 7.3. *If and only if the agenda is non-simple, no aggregation rule generates consistent collective judgment sets and satisfies Universal Domain and Unanimity with Abstentions.*

Proof. First assume the agenda is non-simple. Suppose not for a contradiction that there is an aggregation rule F satisfying universal domain and unanimity with abstentions. We next show that F generates an inconsistent collective judgment set on some profile. By assumption that the agenda is non-simple, then there is a minimally inconsistent set $Y \subseteq X$ with $|Y| \geq 3$. Let α, β, γ be three distinct propositions in Y. Consider a profile $\langle \Phi_i \rangle_{i \in N}$ such that $\Phi_1 = Y \backslash \{\beta\}$, $\Phi_2 = Y \backslash \{\alpha\}$ and $\Phi_i = Y \backslash \{\gamma\}$ for any $i \in N \backslash \{1, 2\}$. Then $\langle \Phi_i \rangle_{i \in N} \in Dom(F)$. And by Unanimity of Abstention, $Y \subseteq F(\langle \Phi_i \rangle_{i \in N})$, so $F(\langle \Phi_i \rangle_{i \in N})$ is inconsistent.

Now assume the agenda is simple. Then there is no minimally inconsistent set $Y \subseteq X$ with $|Y| \geq 3$. Let F be the aggregation rule with universal domain as follows: for all $\langle\Phi_i\rangle_{i\in N} \in Dom(F)$ and for all $\varphi \in X$,

$$\varphi \in F(\langle\Phi_i\rangle_{i\in N}) \text{ iff } \forall_{i\in N}(\neg\varphi \notin \Phi_i) \text{ and } \exists_{x\in N}(\varphi \in \Phi_x)$$

We next prove that F satisfies all requirements.

To show that F satisfies Unanimity with Abstentions, we assume for any $\varphi \in X$, there is some $V \subseteq N$ such that $V \neq \emptyset$, $\forall i \in V.\ \varphi \in \Phi_i$ and $\forall j \in N\backslash V.\ \varphi\#\Phi_j$, then $\neg\varphi \notin \Phi_i$ for any $i \in N$. And by $V \neq \emptyset$, there is $x \in V$ such that $\varphi \in \Phi_x$. So it follows from the definition of F that $\varphi \in F(\langle\Phi_i\rangle_{i\in N})$.

Finally, we consider any profile $\langle\Phi_i\rangle_{i\in N} \in Dom(F)$ and show that $F(\langle\Phi_i\rangle_{i\in N})$ is consistent. Suppose for a contradiction that $F(\langle\Phi_i\rangle_{i\in N})$ is inconsistent for some profile $\langle\Phi_i\rangle_{i\in N} \in Dom(F)$. As $F(\langle\Phi_i\rangle_{i\in N}) \subseteq X$ and by assumption that X is simple, then there is some $p \in X_0$ such that $p \in F(\langle\Phi_i\rangle_{i\in N})$ and $\neg p \in F(\langle\Phi_i\rangle_{i\in N})$. According to the definition of F, $(\forall_{i\in N}(\neg p \notin \Phi_i)$ and $\exists_{x\in N}(p \in \Phi_x))$, and $(\forall_{i\in N}(p \notin \Phi_i)$ and $\exists_{y\in N}(\neg p \in \Phi_y))$, contradiction. Hence, $F(\langle\Phi_i\rangle_{i\in N})$ is consistent for any profile $\langle\Phi_i\rangle_{i\in N} \in Dom(F)$. □

On the one hand, this impossibility result indicates that in decision process when the agenda is logically connected, ignoring abstentions would lead to inconsistent collective results. In fact, unanimity with abstentions means that an individual has the right to determine the collective judgment on certain propositions whenever all the others abstain from voting on that proposition. Thus, to some extent, this impossibility result is a version of Dietrich and List's liberal impossibility result in terms of abstentions. On the other hand, this result provides a characterization theorem for the class of non-simple agendas. It shows that for the class of non-simple agendas, a combination of conditions universal domain, collective consistence and unanimity with abstentions leads to an empty class of aggregation rules. And it also fully characterizes those agendas for which this is the case and, by implication, those for which it is not.

7.2.2.2 Conditions on Aggregation Rules

We now turn to investigating plausible conditions under which the impossibility does not arise. To start with, the following condition is a variant of Unanimity with Abstention by restricting propositions to the literals.

Literal Unanimity with Abstentions (LU). *For every $\alpha \in \mathcal{P}$, if there is some $V \subseteq N$ such that $V \neq \emptyset$, for all $i \in V$. $\alpha \in \Phi_i$ and for all $j \in N \backslash V$. $\alpha \# \Phi_j$, then $\alpha \in F(\langle \Phi_i \rangle_{i \in N})$.*

This condition is a restriction of Unanimity with Abstentions. As we will see in the following example, it plays a crucial role in extending the agenda set from a set of literals to a set of logically interconnected formulas without generating inconsistent aggregate results. It is worth noting that LU is neither a restriction nor an extension of Pareto optimality. The *non-dictatorship* can be still derived from it.

Proposition 7.4. *Every judgment aggregation rule satisfying literal unanimity with abstentions is non-dictatorial.*

The next condition requires that the group judgment on each literal should depend only on individual judgments on that literal, which is a restricted counterpart of Arrow's "independence of irrelevant alternative" [Arrow, 1963].

Literal Independence (LI) . *For every $\alpha \in \mathcal{P}$ and all profiles $\langle \Phi_i \rangle_{i \in N}$, $\langle \Phi'_i \rangle_{i \in N}$ $\in \mathrm{Dom}(F)$, if $\alpha \in \Phi_i \leftrightarrow \alpha \in \Phi'_i$ and $\neg\alpha \in \Phi_i \leftrightarrow \neg\alpha \in \Phi'_i$ for every $i \in N$, then $\alpha \in F(\langle \Phi_i \rangle_{i \in N}) \leftrightarrow \alpha \in F(\langle \Phi'_i \rangle_{i \in N})$.*

This condition amounts to reserving *the independent of irrelevant alternatives condition* in [Dokow and Holzman, 2010] to literals. The following condition is a counterpart of the *neutrality* condition, which requires that an aggregation rule should treat literals in an even-handed way.

Literal Neutrality (LN). For all $\alpha, \beta \in \mathcal{P}$ and every profile $\langle \Phi_i \rangle_{i \in N} \in Dom(F)$, if $\alpha \in \Phi_i \leftrightarrow \beta \in \Phi_i$ and $\neg\alpha \in \Phi_i \leftrightarrow \neg\beta \in \Phi_i$ for every $i \in N$, then $\alpha \in F(\langle \Phi_i \rangle_{i \in N}) \leftrightarrow \beta \in F(\langle \Phi_i \rangle_{i \in N})$.

The last condition is the counterpart of *Systematicity* introduced in [List and Pettit, 2002], which combines independency and neutrality.

Literal Systematicity (LS). *For all $\alpha, \beta \in \mathcal{P}$ and all profiles $\langle \Phi_i \rangle_{i \in N}$, $\langle \Phi'_i \rangle_{i \in N}$ $\in \mathrm{Dom}(F)$, if for every $i \in N$, $\alpha \in \Phi_i \leftrightarrow \beta \in \Phi'_i$ and $\neg\alpha \in \Phi_i \leftrightarrow \neg\beta \in \Phi'_i$, then $\alpha \in F(\langle \Phi_i \rangle_{i \in N}) \leftrightarrow \beta \in F((\Phi'_i)_{i \in N})$.*

The reason why we reserve Independence, Neutrality and Systematicity to literals is that the problem of the doctrinal paradox in Section 1 comes from the requirement that the majority rule treats the compound formulas and propositional variables independently. Indeed *the principle of compositionality*, a fundamental presupposition of the semantics in most contemporary logics, denotes that the propositional variables are more primary than the compound formulas, since the truth of the later is determined by the truth of the former. For instance, in the doctrinal paradox, the truth of the conjunctive formula $p \wedge q$ is determined by its constituents p and q. In this sense, we may say the judgments on p and q are the reasons to accept $p \wedge q$ or not, while the reason for whether p or q is accepted or not is beyond the expressivity of propositional logic.

Therefore, we take a reason-based perspective and apply the aggregation rule only to primary data whose reasons are beyond the expressivity power of the underlying logics, then use them to generate complex formulas within the underlying logic [Mongin, 2008, Nehring and Puppe, 2008]. Given abstentions, it is the literals instead of propositional variables that are primary data. Without completeness, we can not derive that p is rejected from that p is not accepted. It might be possible that p is undetermined (neither accepted nor rejected). Therefore, we reserve Independence, Neutrality and Systematicity to literals instead of propositional variables. On the one hand, this makes them more acceptable. For instance, one criticizes Systematicity (the independent part) being used for $p \vee q$, where p denotes "The government can afford a budget deficit", and q "Forbidding smoking should be legalized" on the ground that there are two propositions involved, and that the society should know how each individual feels about both propositions, and not just about their disjunction. There is no similar objection arising when Systematicity applies to either p or q [Mongin, 2008]. On the other hand, this provides a plausible solution for the paradox shown in Section 1. Let us apply the majority rule to literals and calculate p, q in the group judgment set, then use them to generate $p \wedge q$ in the group judgment set. Thereby, the group judgment set is $\{p, q, r\}$ which is logically consistent with the legal doctrine.

7.3 Aggregation Rule under Voters' Hierarchy

In this section, we first propose a feasible rule for hierarchical groups based on the lexicographic rule in [Andréka et al., 2002], then show that the rule satisfies the plausible conditions in Section 7.2.2.2 and has a tractable computational complexity.

7.3.1 The Literal-based Lexicographic Rule

The priority over individuals is treated as a hierarchy among individuals. In the real-world we can easily see such a hierarchy, for instance, the management structure of an enterprise, a democratic political regime or a community organisation.

Members in different ranks may play different roles in decision-making.

Definition 7.1. A *hierarchy* over set N of individuals is a strict partial order $< \subseteq N \times N$, i.e., it is transitive and asymmetric.

Since N is finite, so there must be no infinite ascending sequence $i_1 < i_2 < i_3 < \cdots$, where $i_n \in N$. In this sense, we call $(N, <)$ is *well-prioritized.* This means all hierarchical chains of N must be "up-bounded" with at least one top leader.

An aggregation rule determines which propositions are collectively accepted and which ones are collectively rejected. Recall that $X_0 = X \cap \mathcal{P}$ is the set of literals in the agenda. We first define an aggregate procedure F for that a literal $\alpha \in X_0$ *is collectively accepted*, denoted by $\alpha \in F(\langle \Phi_i \rangle_{i \in N})$, as follows:

Definition 7.2. For every $\alpha \in X_0$,

$$\alpha \in F(\langle \Phi_i \rangle_{i \in N}) \text{ iff } \forall i \in N(\neg\alpha \notin \Phi_i \vee \exists j \in N(i < j \wedge \alpha \in \Phi_j)) \text{ and } \exists k \in N. \alpha \in \Phi_k. \tag{7.1}$$

Intuitively, this aggregate procedure says that a literal α is accepted by a group if the following two conditions are both satisfied. (i) for any individual if she rejects α, then there is an individual with higher hierarchy accepting α, and (ii) At least one individual accepts α. The set of collectively accepted literals is denoted by $F(\langle \Phi_i \rangle_{i \in N})_0$. Based on this concept, we next define that any formula in the agenda is collectively accepted as follows:

Definition 7.3. For any $\varphi \in X$,

$$\varphi \in F(\langle \Phi_i \rangle_{i \in N}) \text{ iff } F(\langle \Phi_i \rangle_{i \in N})_0 \models \varphi. \tag{7.2}$$

This definition says that a proposition φ in the agenda X is collectively accepted if it is a logical consequence of the collectively accepted literals.

Similarly, a proposition $\varphi \in X$ is *collectively undetermined* if neither itself nor its negation is collectively accepted. That is,

$$\varphi \# F(\langle\Phi_i\rangle_{i\in N}) \text{ iff } \varphi \notin F(\langle\Phi_i\rangle_{i\in N}) \text{ and } \neg\varphi \notin F(\langle\Phi_i\rangle_{i\in N}). \tag{7.3}$$

We call above judgment aggregation rule F *the literal-based lexicographic (judgment aggregation) rule*, since we just apply the lexicographic rule to the subset of literals in the agenda.

To demonstrate how this aggregation rule works, let us consider the following example obtained by a slight change of Example 7.1.

Example 7.2. *Suppose Ann, Bill and Tom have to make group judgments the same three logically connected propositions p, q and $p \to q$. Their individual judgments are shown in Figure 7.4, and the hierarchy among them is illustrated in Figure 7.1. Note that individuals with the highest priority are written at the top of the diagram.*

	p	q	p→q
Ann	+	#	#
Bill	#	−	#
Tom	#	+	+

TABLE 7.4: Individual judgments in Example 7.2.

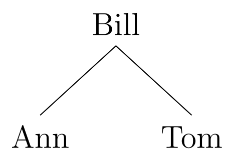

FIGURE 7.1: The hierarchy of Example 7.2.

We now use the literal-based lexicographic rule to generate the collective judgment set. The model of this aggregation situation is given as follows:

- $N = \{Ann, Bill, Tom\}$ with Ann< Bill and Tom< Bill;
- $X = \{p, q, p \to q, \neg p, \neg q, \neg(p \to q)\}$ and $X_0 = \{p, q, \neg p, \neg q\}$.

Let us first calculate the group judgments on the set X_0 of literals according to Definition 7.2.

- The group accepts p, since none of them rejects p, i.e., $\forall i \in N(\neg p \notin \Phi_i)$ holds, and Ann accepts p, i.e., $\exists j \in N(p \in \Phi_j)$ holds.
- The group rejects q since Bill with the highest priority rejects q.

Then the group accepts p and rejects q, i.e., $F(\langle\Phi_i\rangle_{i\in N})_0 = \{p, \neg q\}$. And by Definition 7.3, the group rejects $(p \to q)$, since $\{p, \neg q\} \models \neg(p \to q)$. Thus, the group judgment set is $\{p, \neg q, \neg(p \to q)\}$.

It should be noted that according to the literal-based lexicographic rule, the collective result for Example 7.1 is $\{p, \neg q, \neg(p \to q)\}$, which is consistent. Moreover, this rule also provides a solution to the discursive dilemma in the Background. We may take all the possible hierarchy among the three agents into consideration. One boss case: let $1 < 2 < 3$ be the hierarchy, then according to this rule the aggregate result is just the first individual's judgement set $\{p, q, r\}$. The consistence is obtained at the cost that the first agent seems to be the dictator for this profile. Two-boss case: let the hierarchy be $3 < 1$ and $3 < 2$, then they collectively accept p and abstain from voting on q and r, i.e., $\{p\}$. Three-boss (no boss or anonymity) case: p, q and r are all collectively undetermined.

7.3.2 A Possibility Result

We next show that the literal-based lexicographic rule F is a feasible aggregation rule to generate group judgments, as it satisfies the plausible conditions in Section 7.2.2.2.

Theorem 7.5. *The literal-based lexicographic rule F generates consistent and deductively closed collective judgment sets, and satisfies conditions UD, LU and LS.*

Proof. Regarding consistence, it suffices to show that for any $\alpha \in \mathcal{P}$, $\alpha \in F(\langle\Phi_i\rangle_{i\in N})_0$ implies $\neg\alpha \notin F(\langle\Phi_i\rangle_{i\in N})_0$. Suppose for a contradiction that for some $\beta \in \mathcal{P}$, $\beta \in$

$F(\langle\Phi_i\rangle_{i\in N})_0$ and $\neg\beta \in F(\langle\Phi_i\rangle_{i\in N})_0$, then (i) $\forall_{i\in N}(\neg\beta \notin \Phi_i \vee \exists_{j\in N}(i < j \wedge \beta \in \Phi_j))$ and $\exists_{k\in N}(\beta \in \Phi_k)$; (ii) $\forall_{i\in N}(\beta \notin \Phi_i \vee \exists_{j\in N}(i < j \wedge \neg\beta \in \Phi_j))$ and $\exists_{k\in N}$ $(\neg\beta \in \Phi_k)$. By (i), (ii) we can get an infinite ascending sequence $i_1, i_2, i_3, \cdots$, which is a contradiction with that $(N, <)$ is well-prioritized. Then $F(\langle\Phi_i\rangle_{i\in N})_0$ is consistent, so by Definition 7.3, $F(\langle\Phi_i\rangle_{i\in N})$ is consistent as well.

Regarding deductive closure, for any $\varphi \in X$ assume $F(\langle\Phi_i\rangle_{i\in N}) \models \varphi$, then φ is either a literal or a compound formula. If φ is a literal, then $\varphi \in F(\langle\Phi_i\rangle_{i\in N})_0$, so $\varphi \in F(\langle\Phi_i\rangle_{i\in N})$. If φ is a compound formula, it is straightforward by Definition 7.3. And it is easy to see that F satisfies condition UD.

Regarding LU, assume for every $\alpha \in \mathcal{P}$, if there is some $V \subseteq N$ such that $V \neq \emptyset$, $\forall_{i\in V}$ $(\alpha \in \Phi_i)$ and $\forall_{j\in N\setminus V}(\alpha \# \Phi_j)$, then by Definition 7.2, $\alpha \in F(\langle\Phi_i\rangle_{i\in N})$.

Regarding LS, given every $\alpha \in \mathcal{P}$, the individuals accepting α and these rejecting α are the same for every two profiles $\langle\Phi_i\rangle_{i\in N}$, $\langle\Phi'_i\rangle_{i\in N}$, then the aggregate results of α according to Definition 7.2 are the same as well. Yet it is not the case for LS^s. Consider a counter-example: Let $N = \{1, 2, 3\}$ with $1 < 2$, $1 < 3$. For the profile $\langle\Phi_i\rangle_{i\in N}$ where $\alpha \in \Phi_1$, $\alpha\#\Phi_2$ and $\alpha\#\Phi_3$, we have $\alpha \in F\langle\Phi_i\rangle_{i\in N}$ by LU. Let individuals 2 and 3 who abstain from voting on it turn to rejecting α, while individual 1 still accepts α, we get a different profile $\langle\Phi'_i\rangle_{i\in N}$, where $\alpha \in \Phi'_1$, $\neg\alpha \in \Phi'_2$ and $\neg\alpha \in \Phi'_3$, then $\alpha \in F\langle\Phi'_i\rangle_{i\in N}$ by LS^s, but according to the rule, $\alpha \notin F\langle\Phi'_i\rangle_{i\in N}$. □

It follows directly that the impossibility result of Theorem 7.3 is avoided.

Corollary 7.6. *There exists an aggregation rule that generates consistent collective judgment sets and satisfies conditions UD, LU and LS.*

7.3.3 Computational Complexity

Let us now investigate the computational complexity of winner determination for the literal-based lexicographic rule [Endriss et al., 2012], i.e., how hard is it for this rule to compute the result of a given profile of individual judgement sets. Formally, the decision problem of winner determination is formulated as follows: *for a given formula $\varphi \in X$ and a given profile $\langle \Phi_i \rangle_{i \in N}$, determine whether or not $\varphi \in F(\langle \Phi_i \rangle_{i \in N})$.*

Proposition 7.7. *The winner determination for the literal-based lexicographic rule is in PTIME.*

Proof. On the one hand, we use Algorithm 7.3.1 to compute the set of collective judgments for literals. It is not difficult to check that Algorithm 7.3.1 can be implemented in time $O(m \times n^2)$, where m is the cardinality of X_0 and n is the number of agents. On the other hand, deciding whether a given proposition is accepted by The group amounts to a model-checking problem, which is proved to be in ALOGTIME [Buss, 1987]. Note that ALOGTIME is a complexity class which is subsumed by PTIME. Combining it with the previous result, we then obtain the desired bound. □

Algorithm 7.3.1: Determining Collective Judgments for Literals

Input : A set of literals X_0 and a profile $\langle \Phi_i \rangle_{i \in N}$
Output: A set of literals Y

$Y \leftarrow \emptyset$;
for $\alpha \in X_0$ **do**
    if *for all $i \in N$, $\neg\alpha \notin \Phi_i$ or there is $j \in N$ with $i < j$ such that $\alpha \in \Phi_j$, and there is $k \in N$ such that $\alpha \in \Phi_k$* **then**
        $Y \leftarrow Y \cup \{\alpha\}$
    end
end

7.4 Multiple-Level Collective Decision-Making

In this section we first show that the literal-based lexicographic rule is not oligarchic in the standard sense, and then demonstrate that with abstentions, oligarchic aggregation is not necessarily a single level determination, but can be a multiple-level collective decision-making.

7.4.1 The Oligarchic Property

One may be surprised to find that as an 'unfair' aggregation rule, F is non-dictatorial by Proposition 7.4. This indicates that with abstentions, *non-dictatorship* is a very weak condition imposed on judgment aggregation rules. In fact, it has been shown that by giving up completeness, oligarchies instead of dictatorships are obtained [Dietrich and List, 2008a, Gärdenfors, 2006]. We now investigate in this setting whether this proposed rule is oligarchic. To this end, let us first provide the definition of an oligarchic rule borrowed from [Dietrich and List, 2008a].

Definition 7.4. An aggregation rule G satisfying UD is a weak oligarchy if there is a nonempty smallest subset $M \subseteq N$ such that for every profile $\langle \Phi_i \rangle_{i \in N} \in Dom(G)$,

$$\bigcap_{i \in M} \Phi_i \subseteq G(\langle \Phi_i \rangle_{i \in N}).$$

And an oligarchic rule G is strict if for every profile $\langle \Phi_i \rangle_{i \in N} \in Dom(G)$,

$$\bigcap_{i \in M} \Phi_i = G(\langle \Phi_i \rangle_{i \in N}).$$

In the first (second) case, we call G to be weakly (strictly) oligarchic w.r.t. M.

Special cases of weakly oligarchic aggregation rules are unanimous ($M = N$) and dictatorial ($M = \{i\}$) rules. More specifically, they are both weakly and strictly oligarchic. However, the literal-based lexicographic rule F is neither weakly oligarchic nor strictly oligarchic. Here is a simple counter-example.

Example 7.3. *Let $N = \{1, 2\}$ with $<= \emptyset$, $X = \{p, q, p \to q, \neg p, \neg q, \neg(p \to q)\}$ and $X_0 = \{p, q, \neg p, \neg q\}$, Individual judgment set for each agent is given as follows: $\Phi_1 = \{p, q, p \to q\}$ and $\Phi_2 = \{\neg p, \neg q, p \to q\}$. Then $\Phi_1 \cap \Phi_2 = \{p \to q\}$, but according to the literal-based aggregation rule F, we have that $p \# F(\langle \Phi_1, \Phi_2 \rangle)$, $q \# F(\langle \Phi_1, \Phi_2 \rangle)$ and $p \to q \# F(\langle \Phi_1, \Phi_2 \rangle)$. Thus, $\Phi_1 \cap \Phi_2 \nsubseteq F(\langle \Phi_1, \Phi_2 \rangle)$.*

It should be noted that this does not violates the results in [Dietrich and List, 2008a, Gärdenfors, 2006], since their conditions imposed on aggregation rules are more strengthened than ours: their unanimity and systemacity conditions hold for all formulas, while we restrict them to literals. Therefore, instead of the whole agenda, we need to consider the oligarchy notion with respect to the set of literals in the agenda. This idea leads to a weaker concept of oligarchy as follows:

Definition 7.5. An aggregation rule G satisfying UD is weakly oligarchic w.r.t. X_0 if there is a nonempty smallest subset $M \subseteq N$ such that for every profile $\langle \Phi_i \rangle_{i \in N} \in Dom(G)$,

$$\{\varphi \in X \mid \bigcap_{i \in M} \Phi_i \cap X_0 \models \varphi\} \subseteq G(\langle \Phi_i \rangle_{i \in N}).$$

And an oligarchic rule G is strict w.r.t. X_0 if for every profile $\langle \Phi_i \rangle_{i \in N} \in Dom(G)$,

$$\{\varphi \in X \mid \bigcap_{i \in M} \Phi_i \cap X_0 \models \varphi\} = G(\langle \Phi_i \rangle_{i \in N}).$$

This definition says that an aggregation rule satisfying universal domain is said to be weakly oligarchic w.r.t. X_0, if there is a nonempty smallest set M such that for any profile of individual judgment sets, the group judgment set contains all the consequences of literals that are in every member's judgment set of M. Similarly, an aggregation rule is strictly oligarchic with respect to X_0, if for any profile of individual judgment sets, the group judgment set is exactly the set of consequences of the literals that are in every member's judgment set of M. With this notion, we have the following proposition.

Proposition 7.8. *The literal-based lexicographic rule F is weakly oligarchic w.r.t. X_0, but not strictly oligarchic w.r.t. X_0.*

Proof. Let $O = Max_{\geq}(N) = \{i \in N : \nexists j \in N.i < j\}$. Since $(N, <)$ is well-prioritized and $|N| \geq 2$, so O must exist and cannot be empty. Suppose for every profile of individual judgment sets $\langle\Phi_i\rangle_{i\in N}$, every $i \in N$ and for all $\alpha \in X_0$, $\alpha \in \bigcap_{i\in O}\Phi_i$, then according to Definition 7.2, $\alpha \in F(\langle\Phi_i\rangle_{i\in N})$. Thus, $\bigcap_{i\in O}\Phi_i \cap X_0 \subseteq F(\langle\Phi_i\rangle_{i\in N})_0$. Since $\bigcap_{i\in O}\Phi_i \cap X_0 \models \varphi$, so $F(\langle\Phi_i\rangle_{i\in N})_0 \models \varphi$. And by Definition 7.3, so $\varphi \in F(\langle\Phi_i\rangle_{i\in N})$.

We next show O is the smallest one with $\{\varphi \in X \mid \bigcap_{i\in O}\Phi_i \cap X_0 \models \varphi\} \subseteq F(\langle\Phi_i\rangle_{i\in N})$. Suppose not, then there is some $A \subseteq N$ such that $A \subset O$ and $\{\varphi \in X \mid \bigcap_{i\in A}\Phi_i \cap X_0 \models \varphi\} \subseteq F(\langle\Phi_i\rangle_{i\in N})$, so there is some $a \in N$ such that $a \in O$ but $a \notin A$. Take some $\beta \in X_0$, and define $\beta \in \Phi_i$ for every $i \in N\backslash\{a\}$ and $\neg\beta \in \Phi_a$. Then by Definition 7.2, $\beta \notin F(\langle\Phi_i\rangle_{i\in N})$, but $\beta \in \{\varphi \in X \mid \bigcap_{i\in A}\Phi_i \cap X_0 \models \varphi\}$, contradicting with assumption. Thus, F is weakly oligarchic w.r.t. X_0.

Take $\alpha \in X_0$, and define $\alpha\#\Phi_a$ for some $a \in M$ and $\alpha \in \Phi_x$ for every $x \in N\backslash\{a\}$. By literal unanimity with abstentions, $\alpha \in F(\langle\Phi_i\rangle_{i\in N})$, but $\alpha \notin \bigcap_{i\in M}\Phi_i \cap X_0$. Thus, F is not strictly oligarchic w.r.t. X_0. □

7.4.2 Multi-Level Collective Decision-Making

We next show that with abstentions oligarchic aggregation is not necessarily a single level determination, but can be a multiple-level collective decision making. Before presenting the formal result, we need the following notion. We say that a set of agents D is *decisive* for a judgment aggregation function G if for every profile $\langle \Phi_i \rangle_{i \in N} \in Dom(G)$, $\bigcap_{j \in D} \Phi_j \subseteq G(\langle \Phi_i \rangle_{i \in N})$. Note that a weakly (strictly) oligarchic set is a decisive set, but not the converse, since a decisive set may not be the smallest. We further restrict a decisive set to a specific literal $\alpha \in X_0$ as follows:

Definition 7.6. A set of agents D is decisive on $\alpha \in X_0$ for a judgment aggregation rule G iff for every profile $\langle \Phi_i \rangle_{i \in N} \in Dom(G)$, if $\alpha \in \Phi_j$ for every $j \in D$, then $\alpha \in G(\langle \Phi_i \rangle_{i \in N})$.

It follows that the set $Max_{\geq}(N) = \{i \in N : \nexists j \in N.\ i < j\}$ is decisive for G, yet the decisive set for G is not unique, such as N. Next we partition the hierarchical group N by classifying individuals with the same level into one subgroup from top to bottom according to $<$.

Definition 7.7. Let $<$ be a hierarchy on N. Then induced by $<$, N can be partitioned into subgroups $M_1, \cdots, M_n$, where $\emptyset \neq M_i \subseteq N$ for every $i \in N$, $\bigcup_{i=1}^{n} M_i = N$ and M_i is inductively defined as follows:

- $M_1 = \{i \in N : \nexists j \in N.\ i < j\}$
- $M_{k+1} = \{i \in N \backslash (\bigcup_{i=1}^{k} M_i) : \nexists j \in N \backslash (\bigcup_{i=1}^{k} M_i).\ i < j\}$

It is clear that for every $l, k \in \{1, \cdots, n\}$, if $l \neq k$, then $M_l \cap M_k = \emptyset$. Thus we can say that this is a partition of N in terms of the hierarchical levels. If the group N can be partitioned into n subgroups by above definition, we say the height of the hierarchical group N is n, denoted by $h(N) = n$. For every $i \in N$, if $i \in M_k$, we say the rank of individual i is k. For example, if $a \in M_1$, then a is at the level 1, i.e., at the top of the hierarchy.

The following result displays that none of the superiors rejecting a literal is sufficient and necessary to make the subgroup composed of the immediate inferiors a decisive set on this literal.

Proposition 7.9. *Given a hierarchy among N with $h(N) = n$, let $M_1, \cdots, M_n$ be the subgroups of each level. Then for every $k \in \{0, \cdots, n-1\}$ and $\alpha \in X_0$, M_{k+1} is decisive on α for the literal-based lexicographic rule F if, and only if $\neg\alpha \notin \Phi_i$ for every $i \in \bigcup_{h=0}^{k} M_h$ ($M_0 = \emptyset$).*

Proof. It suffices to prove this result by induction on k. Then

- For $k = 0$, it is trivial as $M_1 = \{i \in N : \nexists j \in N. i < j\}$ is decisive on α for F and $\bigcup M_0 = \emptyset$.
- For $k = l + 1$, where $0 \leq l \leq n - 1$, suppose $\neg\alpha \notin \Phi_i$ for every $i \in \bigcup_{h=0}^{l+1} M_h$, and given arbitrary profile $\langle\Phi_i\rangle_{i\in N} \in Dom(F)$, $\alpha \in \Phi_j$ for every $j \in M_{l+2}$, we need to show $\alpha \in F(\langle\Phi_i\rangle_{i\in N})$. Further assume $\neg\alpha \in \Phi_m$ for every $m \in N$, then $m \notin \bigcup_{h=0}^{l+2} M_h$, according to the definition of subgroups, there must be some superior j for m in M_{l+2} with $\alpha \in \Phi_j$, thus by the definition of F, $\alpha \in F(\langle\Phi_i\rangle_{i\in N})$.

  Conversely, suppose M_{l+2} is decisive on α for F, but there is some $a \in \bigcup_{h=0}^{l+1} M_h$ such that $\neg\alpha \in \Phi_a$. Define a profile $\langle\Phi_i\rangle_{i\in N}$ such that $\neg\alpha \in \Phi_a$, $\alpha \in \Phi_j$ for every $j \in M_{l+2}$ and $\alpha\#\Phi_l$ for every $l \in N \backslash M_{l+2} \backslash \{a\}$. Then $\alpha \notin F(\langle\Phi_i\rangle_{i\in N})$. But from assumption that M_{l+2} is decisive on α for F, and $\alpha \in \Phi_j$ for every $j \in M_{l+2}$, we have $\alpha \in F(\langle\Phi_i\rangle_{i\in N})$: a contradiction.

Thus, the result holds. □

7.5 Discussion and Summary

A growing body of literature on judgement aggregation has emerged in recent years. For an overview of the related research, see [Endriss, 2016, Grossi and Pigozzi, 2014, List, 2012, List and Polak, 2010, List and Puppe, 2009]. In the following, we will review the literature which is most related to our work.

Pioneering work to study expert rights or liberal rights in the context of judgment aggregation is [Dietrich and List, 2008b]. It identifies a liberal paradox for judgment aggregation and explores special conditions to avoid the paradox. Similarly, as a by-product, we obtain an impossibility result: the ignoring of abstentions in decision making may lead to inconsistent collective judgments. It is worth noting that there is no direct relation between the two impossibility results, as we use a hierarchical approach to treat priority over individuals without involving the concept of individual rights in [Dietrich and List, 2008b]. Although the individual with the highest hierarchy (if unique) has the individual rights over the whole agenda, there is no guarantee that two individuals have the rights to decide (at least) one proposition in the agenda. This is why the proposed rule avoids the liberal impossibility.

The idea to employ the lexicographic method for dealing with judgment aggregation under individuals' hierarchy is inspired by [Andréka et al., 2002] where a generalization of the lexicographic rule for combining ordering relations is theoretically studied. In particularly, it applies the lexicographic rule to preference aggregation in social choice and shows that the lexicographic rule is the only way of combining preference relations which satisfies a set of plausible conditions. Different from their motivation, we investigate the collective decision making under a hierarchical environment by proposing a specific procedure based on voters' hierarchy. This may be regarded as an attempt to apply the lexicographic rule for judgment aggregation, which expands the application domain of this rule. We also investigate the computational complexity of this rule which is not involved in [Andréka et al., 2002].

To some extent, the proposed rule may be regarded as a special case of the premiss-based rule [Dietrich and Mongin, 2010, List, 2012]. Dietrich and Mongin [2010] provide a premiss-based approach to judgment aggregation. Their definition of premisses is more general. As a special case, we take the set of literals as the premiss due to the principle of compositionality. Different from our motivation and approaches, they mainly studied necessary and sufficient conditions under which the combination of the premiss-based rule and the conclusion-based rule leads to dictatorship or oligarchy.

In summary, this chapter has provided a hierarchical approach to deal with the priority over individuals and proposed an aggregation rule for judgment aggregation under agents' hierarchy. Meanwhile, we have identified an impossibility result in this setting and explored a set of plausible conditions for aggregation rules in terms of abstentions. We have also shown that the proposed rule satisfies the plausible conditions and has a tractable computational complexity.

Directions of future research are manifold. Firstly, as a special kind of lexicographic rule, it is interesting to investigate a representation result for the proposed rule. The lexicographic rule has been extensively studies in preference aggregation, and [Andréka et al., 2002] has proved that lexicographic rule is the only way of combining preference relations satisfying some natural conditions which are very close to Arrow's conditions. We expect to obtain a similar characterization result. Secondly, under the provision of abstentions, we have investigated a set of commonly desirable conditions. It is natural to investigate some possibility results with respect to these conditions. In addition, with abstentions, the dictatorship in judgment aggregation can also vary in degrees [Rossi et al., 2005]. It is highly interesting to investigate the possibility scope between rationality, dictatorship under a set of plausible conditions. Some work has been done in this direction [Dietrich and List, 2013a].

Chapter 8

Conclusion and Future Work

In this final chapter, I summarize what has been done and discuss what could be done in the future.

8.1 Conclusion

This thesis has proposed a set of logics for modelling strategic reasoning and collective decision-making. Specifically, I have done the following work:

1. GDL-Based Strategic Reasoning

   (a) A logical formalism, called GDR, has been proposed to represent and reason about games strategies. GDR, though looks simple, gains the expressive power from GDL and ATL so that it can be used for describing game rules and game strategies as well as for strategic reasoning. To the best of our knowledge, there is no other logical system that can have the same expressive power with a comparable complexity.

   (b) An epistemic extension of GDL, called EGDL, has been proposed to represent and reason about imperfect information games. EGDL can be used to represent the rules of an imperfect information game, formalize its epistemic properties, and reason about player's epistemic status. Most importantly, the model-checking problem of EGDL is in Δ_2^p, which is the lowest among the existing similar frameworks.

(c) I have further investigated the interplay between knowledge shared by a group of agents and its coalitional abilities by modelling knowledge sharing in ATL, a typical logic for reasoning about coalitional abilities. The relation has been captured through the interplay of epistemic and coalition modalities. Moreover, this semantics is sufficient to preserve the desirable properties of coalitional abilities.

2. Beyond Games: Collective Decision Making

   (a) I have generalized the approach to combine actions via the priority to social choice theory. A modal logic, called RCL, has been proposed by extending propositional logic with the prioritized connective. Not only individual preferences but also aggregation rules can be built into this logic. Thus, individual and collective choices can be automatically generated by a model checking algorithm for RCL.

   (b) I have further provided a logical model for judgment aggregation under voters' hierarchy and designed an aggregation rule based on priorities of individuals so as to investigate how individual judgement affects group judgment in a hierarchical environment. It has been proved that the proposed rule satisfies a set of plausible conditions and has a tractable computational complexity.

Let us stress that, a bottom-up approach was taken to establish the logical formalisms for strategic reasoning in perfect information or imperfect information games: starting with a simple and practical language GDL, and then cautiously extending it so as to retain its practicality. The complexity analysis of these logics indicates that these frameworks make a good balance between expressive power and computational efficiency.

8.2 Future Work

This thesis has taken logic-based approaches to analyze strategic reasoning and collective decision-making. During this process, many new questions arose, some of which have been mentioned in the relevant chapters. I now give four possible directions for future work.

1. The approach to combine actions via the prioritized connectives may be used in the development of general game players. Since the underlying language GDL is the native language for general game playing, with further extension of prioritized strategy connectives, an agent would be able to combine simple actions into more complicated actions. For instance, the simple minmax with alpha-beta heuristics can easily generate the following strategies for mk-games: (1). Fill the center; (2). Fill it if I can win; (3). Fill it if the opponent can win by filling it; (4). Fill one next to mine. Combining them with the prioritized connectives can lead to winning/no-losing strategies for simple mk-games and reasonably good strategies for complicated mk-games. In general, if we do not target winning strategies, automatically discovering strategies with certain desirable properties is possible. In this respect, model-checking is useful for verifying properties of a strategy.

2. In Chapter 4, besides imperfect recall, we have demonstrated that the proposed framework EGDL is flexible enough to specify other memory types, such as state-based memory, action-based memory and perfect recall. To obtain a complete picture of the relation between perfect or imperfect information, and perfect or imperfect recall, we need to study properties of

knowledge with respect to these memory types and investigate the computational complexity in terms of these semantics for knowledge. Moreover, it has been shown that in ATL imperfect information and imperfect recall seems a reasonable compromise, with realistic modelling powers and decidable problems [Schobbens, 2004]. We expect a similar result in EGDL.

3. As in Decision Theory [Peterson, 2009], to make a rational choice in a game is to select the "best" action in light of one's beliefs or information. Thus, to capture epistemic notions in a game situation, besides the knowledge discussed in Chapter 4, we need to enrich the epistemic state transition model with more ingredients such as belief, preference, trust, and further investigate the dynamics of information so as to study the update of players' epistemic attitudes during game play [Liu, 2008, van Benthem et al., 2011, van Ditmarsch et al., 2007]. Some work has been done in this direction. Lorini et al. [2009, 2010] propose some variants of a multi-modal of joint action, preference and knowledge that support reasoning about epistemic games in strategic form. We believe that this thesis provides a good starting point for doing so.

4. The more specific logical properties of GDR and EGDL are worth investigating. Specifically, we want to investigate the satisfiability problem and the axiomatization of GDR and EGDL based on current literature [Goranko and van Drimmelen, 2006, Halpern et al., 2004, Zhang and Thielscher, 2015a]. We also want to compare them with similar existing frameworks such as ATL, ATEL. In particular, Ruan *et al.* build two one-way bridges: one is from GDL to ATL and the other is from GDL-II to ATEL [Ruan and Thielscher, 2012, Ruan et al., 2009]. We expect two similar bridges from GDR to ATL, and from EGDL to ATEL. Moreover, with reasoning facilities, we might explore the converse direction.

Appendix A

Published Work

The following is a list of the published papers during my PhD study.

- Guifei Jiang, Dongmo Zhang, Laurent Perrussel: GDL Meets ATL: A Logic for Game Description and Strategic Reasoning. In Proceedings of the 13th Pacific Rim International Conference on Artificial Intelligence (PRICAI-2014), 733-746, Springer, 2014. *The results of this paper are included in Chapter 2 and Chapter 3.*

- Guifei Jiang, Dongmo Zhang, Laurent Perrussel, Heng Zhang: Epistemic GDL: A logic for representing and reasoning about imperfect information games. Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-2016), 1138-1144, IJCAI/AAAI press, 2016. *The main results of this paper are presented in Chapter 4.*

- Guifei Jiang, Dongmo Zhang, Laurent Perrussel: Knowledge Sharing in Coalitions. In Proceedings of the 28th Australasian Joint Conference on Artificial Intelligence (AI-2015), 249-262, Springer, 2015. *The results in this paper are presented in Chapter 5.*

- Guifei Jiang, Dongmo Zhang, Laurent Perrussel, Heng Zhang: A Logic for Collective Choice. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2015), 979-987, ACM, 2015. *The results in this paper are presented in Chapter 6.*

- Guifei Jiang, Dongmo Zhang, Laurent Perrussel: Judgment Aggregation with Abstentions under Voters' Hierarchy. In Proceedings of the 17th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA-2014), 341-356, Springer, 2014. *The results in this paper are contained in Chapter 7.*

- Guifei Jiang, Dongmo Zhang, Xiaojia Tang: Judgment Aggregation with Abstentions: A Hierarchical Approach. In Proceedings of the 4th International Conference on Logic, Rationality, and Interaction (LORI-2013), 321-325, Springer, 2013. *The results in this paper are contained in Chapter 7.*

- Emiliano Lorini, Guifei Jiang, Laurent Perrussel: Trust-based Belief Change. In Proceedings of the 21th European Conference on Artificial Intelligence (ECAI-2014), 549-554, IOS Press, 2014. *This work is not discussed in the dissertation due to the consideration of length and coherence of the dissertation.*

In addition,

- Guifei Jiang, Dongmo Zhang, Laurent Perrussel: A Hierarchical Approach for Judgment Aggregation with Abstentions. To appear in the Journal of Computational Intelligence, 2016.

Bibliography

Abdou, J. and Keiding, H. (2012). *Effectivity Functions in Social Choice*. Springer Science & Business Media.

Ågotnes, T. (2006). Action and knowledge in alternating-time temporal logic. *Synthese*, 149(2):375–407.

Ågotnes, T., Goranko, V., and Jamroga, W. (2007). Alternating-time temporal logics with irrevocable strategies. In *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'07)*, pages 15–24.

Ågotnes, T., van der Hoek, W., and Wooldridge, M. (2009). Reasoning about coalitional games. *Artificial Intelligence*, 173(1):45–79.

Albert, M. H., Nowakowski, R. J., and Wolfe, D. (2007). *Lessons In Play: An Introduction to Combinatorial Game Theory*. CRC Press.

Alechina, N., Dastani, M., and Logan, B. (2016). Verifying existence of resource-bounded coalition uniform strategies. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*, pages 24–30.

Alur, R., Henzinger, T. A., and Kupferman, O. (2002). Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713.

Andréka, H., Ryan, M., and Schobbens, P.-Y. (2002). Operators and laws for combining preference relations. *Journal of Logic and Computation*, 12(1):13–53.

Arrow, K. (1963). *Social Choice and Individual Values*. Cowles Foundation Monographs Series. Yale University Press.

Arrow, K. J. (1950). A difficulty in the concept of social welfare. *The Journal of Political Economy*, 58(4):328–346.

Bacharach, M. (1994). The epistemic structure of a theory of a game. *Theory and Decision*, 37(1):7–48.

Belardinelli, F. (2014). Reasoning about knowledge and strategies: Epistemic strategy logic. In *Proceedings of the 2nd International Workshop on Strategic Reasoning (SR'14)*, pages 27–33.

Belardinelli, F. (2015). A logic of knowledge and strategies with imperfect information. In *Proceedings of the 8th Workshop on Logical Aspects of Multi-Agent Systems (LAMAS'15)*.

Belle, V. and Lakemeyer, G. (2010). Reasoning about imperfect information games in the epistemic situation calculus. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*, pages 255–260.

Benferhat, S., Cayrol, C., Dubois, D., Lang, J., and Prade, H. (1993). Inconsistency management and prioritized syntax-based entailment. In *Proceedings of the 13th International Joint Conference on Artifical Intelligence (IJCAI'93)*, pages 640–645.

Blackburn, P., De Rijke, M., and Venema, Y. (2002). *Modal Logic*. Cambridge University Press.

Boutilier, C. (1994). Toward a logic for qualitative decision theory. *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, pages 75–86.

Bozzelli, L., Maubert, B., and Pinchinat, S. (2015). Uniform strategies, rational relations and jumping automata. *Information & Computation*, 242:80–107.

Brandt, F., Conitzer, V., and Endriss, U. (2012). Computational social choice. In Weiss, G., editor, *Multiagent Systems*, pages 213–283. MIT Press.

Brewka, G., Benferhat, S., and Le Berre, D. (2004). Qualitative choice logic. *Artificial Intelligence*, 157(1):203–237.

Brihaye, T., Da Costa, A., Laroussinie, F., and Markey, N. (2009). ATL with strategy contexts and bounded memory. In *Proceedings of the International Symposium on Logical Foundations of Computer Science (LFCS'09)*, pages 92–106.

Bulling, N. and Jamroga, W. (2014). Comparing variants of strategic ability: How uncertainty and memory influence general properties of games. *Autonomous Agents and Multi-Agent Systems*, 28(3):474–518.

Bulling, N., Jamroga, W., and Popovici, M. (2014). Agents with truly perfect recall in alternating-time temporal logic. In *Proceedings of the 12th International conference on Autonomous Agents and Multi-Agent Systems (AAMAS'14)*, pages 1561–1562.

Busard, S., Pecheur, C., Qu, H., and Raimondi, F. (2015). Reasoning about memoryless strategies under partial observability and unconditional fairness constraints. *Information and Computation*, 242:128–156.

Buss, S. R. (1987). The boolean formula value problem is in ALOGTIME. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC'87)*, pages 123–131.

Chatterjee, K., Henzinger, T. A., and Piterman, N. (2010). Strategy logic. *Information and Computation*, 208(6):677–693.

Clippel, G. D. and Eliaz, K. (2012). Reason-based choice: A bargaining rationale for the attraction and compromise effects. *Theoretical Economics*, 7:125–162.

Coste-Marquis, S., Lang, J., Liberatore, P., and Marquis, P. (2004). Expressive power and succinctness of propositional languages for preference representation. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'04)*, pages 203–212.

De Giacomo, G., Lespérance, Y., and Pearce, A. R. (2010). Situation calculus based programs for representing and reasoning about game structures. In *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR'10)*, pages 445–455.

Dietrich, F. and List, C. (2007a). Arrow's theorem in judgment aggregation. *Social Choice and Welfare*, 29(1):19–33.

Dietrich, F. and List, C. (2007b). Judgment aggregation by quota rules: Majority voting generalized. *Journal of Theoretical Politics*, 19(4):391–424.

Dietrich, F. and List, C. (2007c). Judgment aggregation with consistency alone. Working Paper, London School of Economics, http://eprints.lse.ac.uk/20110/.

Dietrich, F. and List, C. (2008a). Judgment aggregation without full rationality. *Social Choice and Welfare*, 31(1):15–39.

Dietrich, F. and List, C. (2008b). A liberal paradox for judgment aggregation. *Social Choice and Welfare*, 31(1):59–78.

Dietrich, F. and List, C. (2013a). Propositionwise judgment aggregation: the general case. *Social Choice and Welfare*, 40(4):1067–1095.

Dietrich, F. and List, C. (2013b). A reason-based theory of rational choice. *Nous*, 47(1):104–134.

Dietrich, F. and List, C. (2016). Reason-based choice and context-dependence: An explanatory framework. *Economics and Philosophy*, 32(02):175–229.

Dietrich, F. and Mongin, P. (2010). The premiss-based approach to judgment aggregation. *Journal of Economic Theory*, 145(2):562–582.

Dima, C., Maubert, B., and Pinchinat, S. (2015). Relating paths in transition systems: The fall of the modal mu calculus. In *Proceedings of the 40th International Symposiumon on Mathematical Foundations of Computer Science (MFCS'15)*, pages 179–191.

Dima, C. and Tiplea, F. L. (2011). Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoRR*, abs/1102.4225:http://arxiv.org/abs/1102.4225.

Dokow, E. and Holzman, R. (2010). Aggregation of binary evaluations with abstentions. *Journal of Economic Theory*, 145(2):544–561.

Endriss, U. (2011). Logic and social choice theory. In Gupta, A. and van Benthem, J., editors, *Logic and Philosophy Today*, volume 2, pages 333–377. College Publications.

Endriss, U. (2016). Judgment aggregation. In Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. D., editors, *Handbook of Computational Social Choice*, chapter 17. Cambridge University Press.

Endriss, U., Grandi, U., and Porello, D. (2012). Complexity of judgment aggregation. *Journal of Artificial Intelligence Research*, 45(1):481–514.

Fagin, R. (1996). Combining fuzzy information from multiple systems. In *Proceedings of the 15th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'96)*, pages 216–226.

Fagin, R., Lotem, A., and Naor, M. (2003a). Optimal aggregation algorithms for middleware. *Journal of Computer and System Sciences*, 66(4):614–656.

Fagin, R., Moses, Y., Halpern, J. Y., and Vardi, M. Y. (2003b). *Reasoning about Knowledge*. MIT press.

Gaertner, W. (2009). *A Primer in Social Choice Theory: Revised Edition*. Oxford University Press.

Gärdenfors, P. (2006). A representation theorem for voting with logical consequences. *Economics and Philosophy*, 22(2):181–190.

Gehrlein, W. V. (1983). Condorcet's paradox. *Theory and Decision*, 15(2):161–197.

Gelderie, M. (2012). Strategy machines and their complexity. In *Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science (MFCS'12)*, pages 431–442.

Gelderie, M. (2013). Strategy composition in compositional games. In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP'13)*, pages 263–274.

Gelderie, M. (2014). *Strategy machines: Representation and complexity of strategies in infinite games*. PhD thesis, Department of Computer Science, RWTH Aachen University.

Genesereth, M., Love, N., and Pell, B. (2005). General game playing: Overview of the AAAI competition. *AI Magazine*, 26(2):62–72.

Ghosh, S. (2008). Strategies made explicit in dynamic game logic. In van Benthem, J. and Pacuit, E., editors, *Proceedings of the Workshop on Logic and Intelligent Interaction (ESSLLI'08)*, pages 74–81.

Goranko, V. and van Drimmelen, G. (2006). Complete axiomatization and decidability of alternating-time temporal logic. *Theoretical Computer Science*, 353(1):93–117.

Gottlob, G. (1995). NP trees and Carnap's modal logic. *Journal of the ACM*, 42(2):421–457.

Governatori, G. and Rotolo, A. (2006). Logic of violations: A gentzen systems for reasoning with contrary-to-duty obligations. *Australasian Journal of Logic*, 4(1):193–215.

Grossi, D. and Pigozzi, G. (2014). *Judgment Aggregation: A Primer*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Guelev, D. P., Dima, C., and Enea, C. (2011). An alternating-time temporal logic with knowledge, perfect recall and past: axiomatisation and model-checking. *Journal of Applied Non-Classical Logics*, 21(1):93–131.

Halpern, J. Y., van Der Meyden, R., and Vardi, M. Y. (2004). Complete axiomatizations for reasoning about knowledge and time. *SIAM Journal on Computing*, 33(3):674–703.

Halpern, J. Y. and Vardi, M. Y. (1989). The complexity of reasoning about knowledge and time. I. lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237.

Harel, D., Kozen, D., and Tiuryn, J. (2000). *Dynamic Logic*. MIT press.

Haufe, S. and Thielscher, M. (2012). Automated verification of epistemic properties for general game playing. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR'12)*, pages 339–349.

Herzig, A. (2015). Logics of knowledge and action: critical analysis and challenges. *Autonomous Agents and Multi-Agent Systems*, 29(5):719–753.

Herzig, A. and Lorini, E. (2010). A dynamic logic of agency I: STIT, capabilities and powers. *Journal of Logic, Language and Information*, 19(1):89–121.

Herzig, A., Lorini, E., and Walther, D. (2013). Reasoning about actions meets strategic logics. In *Proceedings of the 4th International Workshop on Logic, Rationality, and Interaction (LORI'13)*, pages 162–175.

Herzig, A. and Troquard, N. (2006). Knowing how to play: uniform choices in logics of agency. In *Proceeding of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06)*, pages 209–216.

Huang, X., Chen, Q., and Su, K. (2016). Strengthening agents strategic ability with communication. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, pages 2509–2515.

Huang, X., Ruan, J., and Thielscher, M. (2013). Model checking for reasoning about incomplete information games. In *Proceedings of the 26th Australasian Joint Conference on Advances in Artificial Intelligence (AI'13)*, pages 246–258.

Huang, X. and van der Meyden, R. (2014a). An epistemic strategy logic. In *Proceedings of the 2nd International Workshop on Strategic Reasoning (SR'14)*, pages 35–41.

Huang, X. and van der Meyden, R. (2014b). A temporal logic of strategic knowledge. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14)*, pages 418–427.

Jamroga, W. and Bulling, N. (2011). Comparing variants of strategic ability. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 252–257.

Jamroga, W. and van der Hoek, W. (2004). Agents that know how to play. *Fundamenta Informaticae*, 63(2):185–219.

Kaneko, M. and Nagashima, T. (1996). Game logic and its applications I. *Studia Logica*, 57(2-3):325–354.

Kaneko, M. and Nagashima, T. (1997). Game logic and its applications II. *Studia Logica*, 58(2):273–303.

Kaźmierczak, P., Ågotnes, T., and Jamroga, W. (2014). Multi-agency is coordination and (limited) communication. In *Proceedings of the 17th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA'14)*, pages 91–106.

Konieczny, S. and Pérez, R. P. (2002). Merging information under constraints: a logical framework. *Journal of Logic and Computation*, 12(5):773–808.

Kornhauser, L. A. and Sage, L. G. (1993). The one and the many: Adjudication in collegial courts. *California Law Review*, 81(1):1–59.

Kuhn, H. W. (1953). Extensive games and the problem of information. *Contributions to the Theory of Games*, 2(28):193–216.

Lafage, C. and Lang, J. (2000). Logical representation of preferences for group decision making. In *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR'00)*, pages 457–470.

Lang, J. (2004). Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence*, 42(1-3):37–71.

Lang, J. (2006). Logical representation of preference: A brief survey. In Riccia, G., Dubois, D., Kruse, R., and Lenz, H.-J., editors, *Decision Theory and Multi-Agent Planning*, pages 65–88. Springer.

List, C. (2012). The theory of judgment aggregation: An introductory review. *Synthese*, 187(1):179–207.

List, C. and Pettit, P. (2002). Aggregating sets of judgments: An impossibility result. *Economics and Philosophy*, 18(01):89–110.

List, C. and Pettit, P. (2004). Aggregating sets of judgments: Two impossibility results compared. *Synthese*, 140(1-2):207–235.

List, C. and Polak, B. (2010). Introduction to judgment aggregation. *Journal of economic theory*, 145(2):441–466.

List, C. and Puppe, C. (2009). Judgment aggregation: A survey. In P. Anand, P. P. and Puppe, C., editors, *The Handbook of Rational and Social Choice*. Chapter 19. Oxford University Press.

Liu, F. (2008). *Logic for Social Software*. PhD thesis, University of Amsterdam.

Lorini, E. and Schwarzentruber, F. (2010). A modal logic of epistemic games. *Games*, 1(4):478–526.

Love, N., Hinrichs, T., Haley, D., Schkufza, E., and Genesereth, M. (2006). General game playing: Game description language specification. Stanford Logic Group Computer Science Department Stanford University. http://logic.stanford.edu/reports/LG-2006-01.pdf.

Maubert, B. and Pinchinat, S. (2013). Jumping Automata for Uniform Strategies. In Seth, A. and Vishnoi, N. K., editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'13)*, volume 24, pages 287–298. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Maubert, B. and Pinchinat, S. (2014). A general notion of uniform strategies. *International Game Theory Review*, 16(1).

Maubert, B., Pinchinat, S., and Bozzelli, L. (2013). The complexity of synthesizing uniform strategies. In *Proceedings of the 1st International Workshop on Strategic Reasoning (SR'13)*, pages 115–122.

May, K. O. (1952). A set of independent necessary and sufficient conditions for simple majority decision. *Econometrica: Journal of the Econometric Society*, pages 680–684.

Mogavero, F. (2013). Reasoning about strategies. In *Proceedings of the 28th Annual ACM/IEEE Symposium on Logics in Computer Science (LICS'13)*, pages 85–116.

Mogavero, F., Murano, A., Perelli, G., and Vardi, M. Y. (2014). Reasoning about strategies: On the model-checking problem. *ACM Transactions on Computational Logic (TOCL)*, 15(4):1–47.

Mogavero, F., Murano, A., and Vardi, M. Y. (2010). Reasoning about strategies. In *Proceedings of the 30th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'10)*, pages 133–144.

Mongin, P. (1995). Consistent bayesian aggregation. *Journal of Economic Theory*, 66(2):313–351.

Mongin, P. (2008). Factoring out the impossibility of logical aggregation. *Journal of Economic Theory*, 141(1):100–113.

Nehring, K. and Puppe, C. (2008). Consistent judgement aggregation: The truth-functional case. *Social Choice and Welfare*, 31(1):41–57.

Nuel, B., Michael, P., and Ming, X. (2001). *Facing the Future: Agents and Choices in Our Indeterminist World*. Oxford University Press.

Osborne, M. J. (1994). *A Course in Game Theory*. MIT Press.

Osherson, D. and Weinstein, S. (2012). Preference based on reasons. *The Review of Symbolic Logic*, 5(01):122–147.

Pacuit, E. (2014). Strategic reasoning in games. Manuscript. http://web.pacuit.org/esslli2012/handouts-stratreas/reas-games-v1.pdf.

Pacuit, E. (2015). On the use (and abuse) of logic in game theory. *Journal of Philosophical Logic*, 44(6):741–753.

Parikh, R. (1985). The logic of games and its applications. *North-Holland Mathematics Studies*, 102:111–139.

Patty, J. W. and Penn, E. M. (2014). *Social Choice and Legitimacy: The Possibilities of Impossibility*. Cambridge University Press.

Pauly, M. (2001). *Logic for Social Software*. PhD thesis, University of Amsterdam.

Pauly, M. (2002). A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166.

Pauly, M. and Parikh, R. (2003). Game logic-An overview. *Studia Logica*, 75(2):165–182.

Pedersen, T., Dyrkolbotn, S., and Ågotnes, T. (2013). Reasonably rational: Reasoning about reasons behind preferences using modal logic. In *Proceedings of the 2nd International Conference on Agreement Technologies (AT'13)*, pages 120–134.

Peleg, B. (1997). Effectivity functions, game forms, games, and rights. *Social choice and welfare*, 15(1):67–80.

Perea, A. (2014). From classical to epistemic game theory. *International Game Theory Review*, 16(01).

Peterson, M. (2009). *An Introduction to Decision Theory*. Cambridge University Press.

Piccione, M. and Rubinstein, A. (1997). On the interpretation of decision problems with imperfect recall. *Games and Economic Behavior*, 20(1):3–24.

Pigozzi, G. (2006). Belief merging and the discursive dilemma: An argument-based account to paradoxes of judgment aggregation. *Synthese*, 152(2):285–298.

Polak, B. (2007). Backward induction: Chess, strategies, and credible threats. http://oyc.yale.edu/economics/econ-159/lecture-15.

Porello, D. and Endriss, U. (2011). Ontology merging as social choice. In *Proceedings of the 12th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA'11)*, pages 157–170.

Pritchard, D. B. (1994). *The Encyclopedia of Chess Variants*. Games & Puzzles.

Ramanujam, R. and Simon, S. (2006). Axioms for composite strategies. In *Proceedings of the 7th conference on Logic and the Foundations of Game and Decision Theory (LOFT'06)*, pages 189–198.

Ramanujam, R. and Simon, S. (2008a). A logical structure for strategies. In *Proceedings of the 8th conference on Logic and the Foundations of Game and Decision Theory (LOFT'08)*, pages 183–208.

Ramanujam, R. and Simon, S. E. (2008b). Dynamic logic on games with structured strategies. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 49–58.

Reiter, R. (1991). The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, 27:359–380.

Rossi, F., Venable, K. B., and Walsh, T. (2005). Aggregating preferences cannot be fair. *Intelligenza Artificiale*, 2(1):30–38.

Ruan, J. and Thielscher, M. (2011). The epistemic logic behind the game description language. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*, pages 840–845.

Ruan, J. and Thielscher, M. (2012). Strategic and epistemic reasoning for the game description language GDL-II. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12)*, pages 696–701.

Ruan, J., van Der Hoek, W., and Wooldridge, M. (2009). Verification of games in the game description language. *Journal of Logic and Computation*, 19(6):1127–1156.

Schiffel, S. and Thielscher, M. (2011). Reasoning about general games described in GDL-II. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*, pages 846–851.

Schiffel, S. and Thielscher, M. (2014). Representing and reasoning about the rules of general games with imperfect information. *Journal of Artificial Intelligence Research*, 49:171–206.

Schobbens, P.-Y. (2004). Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93.

Sen, A. (1970). The impossibility of a paretian liberal. *The Journal of Political Economy*, 78(1):152–157.

Sistla, A. P. and Clarke, E. M. (1985). The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749.

Tan, S.-W. and Pearl, J. (1994). Qualitative decision theory. In *Proceedings of the 12th AAAI Conference on Artificial Intelligence (AAAI'94)*, pages 928–933.

Thielscher, M. (2010). A general game description language for incomplete information games. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*, pages 994–999.

Thielscher, M. (2011). The general game playing description language is universal. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 1107–1112.

Troquard, N. and Walther, D. (2012). On satisfiability in ATL with strategy contexts. In *Proceedings of the 13th European Conference on Logics in Artificial Intelligence (JELIA'12)*, pages 398–410.

Valiev, M. K. (1980). Decision complexity of variants of propositional dynamic logic. In *Proceedings of the 9th Symposium on Mathematical Foundations of Computer Science (MFCS'80)*, pages 656–664.

van Benthem, J. (2001). Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–248.

van Benthem, J. (2002). Extensive games as process models. *Journal of Logic, Language and Information*, 11(3):289–313.

van Benthem, J. (2012). In praise of strategies. In van Eijck, J. and Verbrugge, R., editors, *Games, Actions and Social Software: Multidisciplinary Aspects*, pages 96–116. Springer Berlin Heidelberg.

van Benthem, J. (2013). Reasoning about strategies. In *Computation, Logic, Games, and Quantum Foundations. The Many Facets of Samson Abramsky*, pages 336–347. Springer.

van Benthem, J. (2014). *Logic in games*. MIT Press.

van Benthem, J., Ghosh, S., and Liu, F. (2008). Modelling simultaneous games in dynamic logic. *Synthese*, 165(2):247–268.

van Benthem, J., Girard, P., and Roy, O. (2009). Everything else being equal: A modal logic for ceteris paribus preferences. *Journal of Philosophical Logic*, 38(1):83–125.

van Benthem, J., Grossi, D., and Liu, F. (2014). Priority structures in deontic logic. *Theoria*, 80(2):116–152.

van Benthem, J., Pacuit, E., and Roy, O. (2011). Toward a theory of play: A logical perspective on games and interaction. *Games*, 2(1):52–86.

van den Herik, H. J., Uiterwijk, J. W., and van Rijswijck, J. (2002). Games solved: Now and in the future. *Artificial Intelligence*, 134(1):277–311.

van der Hoek, W., Jamroga, W., and Wooldridge, M. (2005). A logic for strategic reasoning. In *Proceedings of the 3th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05)*, pages 157–164.

van der Hoek, W. and Pauly, M. (2006). Modal logic for games and information. In Blackburn, P., van Benthem, J., and Wolter, F., editors, *Handbook of Modal Logic*, volume 3, pages 1077–1148. Elsevier.

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 Ditmarsch, H., Halpern, J. Y., van der Hoek, W., and Kooi, B. P. (2015). *Handbook of Epistemic Logic*. College Publications.

van Ditmarsch, H. and Knight, S. (2014). Partial information and uniform strategies. In *Proceedings of the 15th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA'14)*, pages 183–198.

van Ditmarsch, H., van der Hoek, W., and Kooi, B. P. (2007). *Dynamic Epistemic Logic*, volume 337. Springer Science & Business Media.

van Eijck, J. (2013). PDL as a multi-agent strategy logic. In *Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'13)*, pages 206–215.

van Hees, M. (1999). Liberalism, efficiency, and stability: some possibility results. *Journal of Economic Theory*, 88(2):294–309.

Walther, D., van der Hoek, W., and Wooldridge, M. (2007). Alternating-time temporal logic with explicit strategies. In *Proceedings of the 11th conference on Theoretical Aspects of Rationality and Knowledge (TARK'07)*, pages 269–278.

Waugh, K., Zinkevich, M., Johanson, M., Kan, M., Schnizlein, D., and Bowling, M. H. (2009). A practical use of imperfect recall. In *Proceedings of the 8th Symposium on Abstraction, Reformulation, and Approximation (SARA'09)*, pages 175–182.

Wilson, R. (1975). On the theory of aggregation. *Journal of Economic Theory*, 10(1):89–99.

Wooldridge, M., Ågotnes, T., Dunne, P. E., and van der Hoek, W. (2007). Logic for automated mechanism design - A progress report. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI'07)*, pages 9–16.

Zhang, D. and Thielscher, M. (2015a). A logic for reasoning about game strategies. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 1671–1677.

Zhang, D. and Thielscher, M. (2015b). Representing and reasoning about game strategies. *Journal of Philosophical Logic*, 44(2):203–236.