Epistemic and Doxastic Planning

Jensen, Martin Holm; Bolander, Thomas

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

Epistemic and Doxastic Planning

Martin Holm Jensen

Kongens Lyngby 2014
PhD-2014-316

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Matematiktorvet, Building 303 B, DK-2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

PhD-2014-316

Abstract

This thesis is concerned with planning and logic, both of which are core areas of Artificial Intelligence (AI). A wide range of research disciplines deal with AI, including philosophy, economy, psychology, neuroscience, mathematics and computer science. The approach of this thesis is based on mathematics and computer science. Planning is the mental capacity that allows us to predict the outcome of our actions, thereby enabling us to exhibit goal-directed behaviour. We often make use of planning when facing new situations, where we cannot rely on entrenched habits, and the capacity to plan is therefore closely related to the reflective system of humans. Logic is the study of reasoning. From certain fixed principles logic enables us to make sound and rational inferences, and as such the discpline is virtually impossible to get around when working with AI.

The basis of automated planning, the term for planning in computer science, is essentially that of propositional logic, one of the most basic logical systems used in formal logic. Our approach is to expand this basis so that it is based on richer and and more expressive logical systems. To this end we work with logics for describing knowledge, beliefs and dynamics, that is, systems that allow us to formally reason about these aspects. By letting these elements be used in a planning context, we obtain a system that extends the degree to which goal-directed behaviour can, at present, be captured by automated planning.

In this thesis we concretely apply dynamic epistemic logic to capture knowledge, and dynamic doxastic logic for capturing belief. We highlight two results of this thesis. The first pertains to how dynamic epistemic logic can be used to describe the (lack of) knowledge of an agent in the midst of planning. This perspective is already incorporated in automated planning, and seen in isolation this result appears mainly as an alternative to existing theory. Our second result underscores the strength of the first. Here we show how the kinship between the aforementioned logics enable us to extend automated planning with doxastic elements. An upshot of expanding the basis of automated planning is therefore that it allows for a modularity, which facilitates the introduction of new aspects

into automated planning.

We round things off by describing what we consider to be the absolutely most fascinating perspective of this work, namely situations involving multiple agents. Reasoning about the knowledge and beliefs of others are essential to act rationally. It enables cooperation, and additionally forms the basis for engaging in a social context. Both logics mentioned above are formalized to deal with multiple agents, and the first steps have been taken towards extending automated planning with this aspect. Unfortunately, the first results in this line of research have shown that planning with multiple agents is computationally intractable, and additional work is therefore necessary in order to identify meaningful and tractable fragments.

Resumé

Denne afhandling beskæftiger sig med planlægning og logik, der begge er kerneområder inden for Kunstig Intelligens (KI). En bred vifte af forskningsdiscipliner behandler KI, herunder bl.a. filosofi, økonomi, psykologi, neurologi, matematik og datalogi. Tilgangen i denne afhandling er funderet i matematikken og datalogien. Planlægning er den mentale evne, som tillader os at forudsige udfaldet af vores handlinger, og herigennem gør os i stand til at udvise målbevidst adfærd. Vi benytter os ofte af planlægning i nye situationer, hvor vi ikke kan forlade os på indgroede vaner, og evnen til at planlægge er derfor knyttet til menneskets refleksive system. Logik er læren om at ræsonnere. Ud fra visse kerneprincipper gør logik os i stand til at lave korrekte og rationelle slutninger, og derfor er disciplinen stort set umulig at komme uden om i arbejdet med KI.

Grundteorien i automatiseret planlægning, den datalogiske betegnelse for planlægning, er essentielt bygget på udsagnslogik, hvilket er et af de simpleste systemer, der benyttes i den formelle logik. Vores fremgangsmåde er at udvide denne grundteori, så den baseres på stærkere og mere udtryksfulde logiske systemer. Vi beskæftiger os i den forbindelse med logikker der beskriver viden, overbevisning og dynamik, dvs. systemer der gør os i stand til formelt at ræsonnere om disse aspekter. Ved at lade disse elementer indgå i en planlægningssammenhæng opnås en udvidelse af dén målbevidste adfærd, der i dag kan beskrives i automatiseret planlægning.

Konkret benytter vi i afhandlingen dynamisk epistemisk logik til at beskrive viden, samt dynamisk doksastisk logik til at beskrive overbevisning. Vi fremhæver to resultater fra afhandlingen. Det første drejer sig om, hvordan dynamisk epistemisk logik kan benyttes til at beskrive en aktørs (manglende) viden mens denne er i gang med planlægge. Dette perspektiv findes allerede inkorporeret i planlægning, og isoleret set fremstår dette resultat hovedsageligt som et alternativ til eksisterende teori. Vores andet resultat fremhæver dog styrken af det første. Her viser vi, hvordan slægtskabet mellem de to føromtalte logikker gør os i stand til at udvide planlægning med doksatiske elementer. En af styrkerne ved

at løfte grundteorien i planlægning er altså en modularitet, som langt lettere tillader indfasning af nye aspekter i automatiseret planlægning.

Vi runder af med at beskrive, hvad vi betragter som det absolut mest fascinerende fremtidsperspektiv for dette arbejde, nemlig situationer der omfatter flere aktører. At kunne ræsonnere om andre aktørers viden og overbevisning er essentielt for at kunne agere rationelt. Det er muliggør samarbejde, og danner derudover grundlaget for at kunne indgå i sociale sammenhænge. Begge logikker nævnt ovenfor er formaliseret til at udtale sig om flere aktører, og de første spadestik er taget til udvidelse af planlægning med dette aspekt. Uheldigvis har de første resultater på dette punkt vist, at planlægning med adskillige aktører er beregningsmæssigt fuldstændig uhåndterbart, og yderligere arbejde er nødvendigt for at kunne isolere meningsfyldte fragmenter.

Preface

The work in thesis is a result my PhD study carried out in the section for Algorithms, Logic and Graphs in the Department of Applied Mathematics and Computer Science, Technical University of Denmark, DTU. The results presented were obtained in the period from September 2010 to February 2014. My supervisor was Associate Professor Thomas Bolander. In the period from September 2011 to December 2011 I was on an external research stay at the University of Amsterdam, under the supervision of Alexandru Baltag and Johan van Benthem.

This thesis consists of three joint publications and two addenda presenting further results. It is part of the formal requirements to obtain the PhD degree at the Technical University of Denmark.

Lyngby, 31-January-2014

Martin Holm Jensen

Acknowledgements

Above all I must thank Thomas Bolander who has been an outstanding supervisor throughout my PhD studies. Thomas has tirelessly guided, advised and educated me during this period, and I hope that this thesis is able to reflect at least some of his profound dedication.

I am delighted to have had close collaboration with Mikkel Birkegaard Andersen in much of the work I have done throughout my PhD studies. I must not neglect to mention all of the other great individuals working in the section for Algorithms, Logic and Graphs at DTU, which have provided for an excellent environment both socially and academically. I name a few of the individuals with whom I've had fruitful discussions with, which have contributed to this thesis: Hans van Ditmarsch, Jens Ulrik Hansen, Bruno Zanuttini, Jérôme Lang, Valentin Goranko, Johan van Benthem and Alexandru Baltag. Additionally, I highly appreciate the very insightful remarks provided by the members of my assessment committee, which consisted of Valentin Goranko, Andreas Herzig and Thomas Ågotnes.

Lastly, I am indebted and grateful to my dearest Louise for putting up with me during the final stages of writing this thesis.

Contents

Chapter 1

Introduction

Artificial intelligence (AI) is a line of research that brings together a vast number of research topics. Indeed, subfields of AI include diverse areas such as philosophy, mathematics, economics, neuroscience, psychology, linguistics and computer science. The ideas and techniques found within this motley crew of disciplines have laid the foundation of AI, and, to this day, still contributes to its further development. We can equate the term "artificial" with something that is man-made, that is, something which is conceived and/or crafted by humans. Pinning down what constitutes "intelligence" is on the other hand a daunting and controversial task, one which we will not undertake here. Instead we will posit that formal reasoning plays a crucial role in designing and building intelligent systems, and that we must require a generality of the reasoning process to such an extent that it can be applied to many different types of problems. This sentiment is our principal motivation for the work conducted in this thesis.

Pertaining more generally to the development of AI, we support the view of Marvin Minksy that we shouldn't look for a "magic bullet" for solving all kinds of problems [Bicks, 2010]. In accordance with this stance we make no claim that the methods presented here are universally best for all tasks. Nevertheless we find that these methods capture relevant and useful aspects of reasoning, as is also evident from studies in cognitive science (see *theory of mind* in Section 1.5).

We bring up two virtually ubiquitous notions in AI. The first is that of an *agent*. To us an agent constitutes some entity that is present, and possibly acts, in an environment. An agent can represent e.g. a human or a robot acting in the world, but in general an agent need not have a physical representation and so

can equally well be a piece of software operating in some environment. We also mention environments in which multiple agents are present, each of which are then taken to represent an individual entity. The second notion is that of *rational agency*, which is admittedly more elusive than the first. Following [van Benthem, 2011] we take rational agency to mean that agents must have some form of rationale behind the choices they make, and that this rationale is based on an agent's knowledge, goals and preferences.

In the remainder of this chapter we paint a picture of the topics of AI we treat in this thesis, and along the way point out relationships with results contained in this thesis. We start out with automated planning in Section 1.1, where we explicate the basic formalism underlying this research discipline. Section 1.2 is a brief account of dynamic epistemic logic, a topic we revisit many times over in later chapters. Following this, Section 1.3 describes what we'll refer to as epistemic planning, where the theory of the two aforementioned areas come together. In Section 1.4 we motivate research into the area of epistemic planning and further discuss a few applications, before we touch upon related formalisms in Section 1.5. We wrap this chapter up in Section 1.6 by outlining the contents and contributions of subsequent chapters.

1.1 Automated Planning

When planning we want to organize our actions so that in a certain situation, our goal, is brought about. This organization of actions can be represented as a plan for how we should act. By predicting the outcome of the actions at our disposal, we have a method for generating plans that lead us to our goal. In the textbook [Ghallab et al., 2004] planning is called the "reasoning side of acting", meaning that it constitutes a process of careful deliberation about the capabilities we're endowed with. The following presentation is mainly based on [Ghallab et al., 2004], though we fill in some blanks using the thorough exposition of planning due to [Rintanen, 2005]. Our aim is to give the reader a feel for the theory underlying automated planning, while keeping notation and long-winded digressions to a minimum.

Our point of departure is what constitutes the simplest form of planning problem. In this case we consider three components: An initial state, a set of actions (an action library) and a goal, with each of these being represented using simple notions from set theory. In this set-theoretic formulation we take a finite set of symbols P. A state is taken to be a subset of P, and we let actions manipulate states by removing and/or adding symbols. A goal is a collection of states (a subset of the powerset of P). The intuition is that a symbol $p \in P$ is true in a

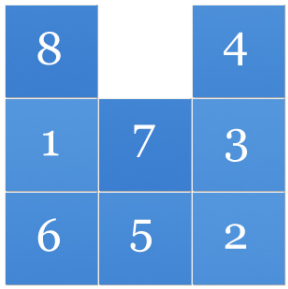


Figure 1.1: Example of the initial state of a sliding puzzle problem.

state s iff $p \in s$. Furthermore, a set of symbols are assigned as the precondition of an action. If each symbol is contained in a state s, the action is applicable meaning that its execution is allowed. A solution to a planning problem is a sequence of actions, each of which is applicable, and which transforms the initial state into one of the goal states.

Example 1.1. To warm up we consider the sliding puzzle problem; see Figure 1.1. The goal here is to have each piece arranged so that the sequence $1, 2, 3, \ldots$ is formed (horizontally, starting from the top left corner). This problem can be formulated as a planning problem, with the location of each numbered piece making up a state; and the initial state is then the location of pieces as in the illustration. There is a single goal state, namely the state in which each piece is arranged correctly. An action is to move a single piece in some direction (up, left, down, right). In this instance we prevent an action such as "Move piece 1 up" from being executed by utilizing applicability. With this a sliding puzzle problem always has between two and four applicable actions, depending on which location is unoccupied. In Example 1.4 we show how this problem can be formulated in a specification language.

The sliding puzzle problem is an example of what can be formulated as a planning problem. Other examples include the blocks-world problem, truly an evergreen, where blocks must be moved around on a table so that a particular arrangement of the blocks is obtained, as well as more exotic problems such as operating a number of satellites that take pictures of cosmological phenomena. The point here is that by working with a generic formulation, automated planning can be applied to a wide range of scenarios, and this generality is what we're after in AI.

There are other formulations than the set-theoretic one given above. One is to consider states to be conjunctions of atoms, and effects (of actions) to be conjunctions of literals with negative literals being removed and positive liter-

als being added. In this formulation we can immediately see a connection to propositional logic, though often the relationship between logic and automated planning is much less evident. The seminal [Fikes and Nilsson, 1971], which introduced the generic problem solver "STRIPS" (STanford Research Institute Problem Solver), used a much more expressive formulation of planning, including the ability to use arbitrary formulas of first-order logic as effects. As it turned out to be hard to give well-defined semantics for this formulation, the compromise based on propositional logic was adopted [Ghallab et al., 2004].

Underlying our presentation in the above are several assumptions about the environment. It is fully observable as a symbol is known to be either true or false; it is deterministic as the execution of an action brings the environment to a single other state; and it is static as only the planning agent brings about change to the environment. The reasoning (or, planning) process is offline meaning that only when the agent acts is the environment modified, and so there is no need for interleaving of planning and execution. The type of planning that falls under these restrictions is often referred to as as classical planning.

Already present in classical planning is the important distinction between plan-time and run-time. The former is when the planning agent is deliberating about what course of action it must take to achieve its goal, and the latter is when a plan is actually executed. This distinction becomes crucial as we now turn to environments with partial observability and nondeterminism, by relaxing some of the assumptions of classical planning. The general formulation in automated planning is based on Partially Observable Markov Decision Processes (POMDPs), meaning that both observations and actions are assigned probability distributions and so this leads to an inherently stochastic environment. As a consequence of this choice of modelling, when treating environments that are nondeterministic and partially observable yet not stochastic, automated planning employs a non-probabilistic form of POMDPs.

Partial observability leads to the notion of a belief state (a set of states; i.e. subset of 2^P), which signifies that what is known is only that one state in a belief state is, or will be, the actual situation.[1] There are various approaches

[1] There is actually an important remark about our use of the future continuous form *will be* here. We use it to underscore that we're in the midst of a reasoning process (plan-time), and so it seems to indicate a future situation that just hasn't happened yet. However, exactly because we're in a deliberation phase, this particular future can only come about should the agent choose to act in a particular way. Even still, due to the nondeterministic nature of the environment it may not be possible for the agent to determine (at plan-time), whether this future situation is going to be realized at all. On top of this, the agent may also be ignorant about certain facts, and so what we're trying to convey here contains several layers of intricacy. Rather than canvassing for a grammatically more accurate statement, we'll stick with the future continuous form and let the precise meaning be deduced from the mathematical framework in question.

to how belief states are induced, that is, when two states are considered indistinguishable to the agent. One method is to use a set of observation variables $V \subseteq P$, denoting exactly the propositions whose truth value can be observed at run-time. These variables are automatically observed after each action, while the truth value of variables in $P \setminus V$ can never be determined by the agent. Nondeterminism is handled by allowing actions to manipulate states in more than one way, so, for instance, an action may or may not add a proposition symbol to a state. One result from this choice of modelling is that observability and nondeterminism are completely detached from one another; actions still manipulate states (not belief states) and observation variables are statically given and independent of actions. We should note that [Rintanen, 2005, Section 4.1.1] shows how this representation is able to express observing the truth of arbitrary propositional formulas over P upon the execution of a particular action (referred to as sensing actions).

Example 1.2. Consider an agent having two dice cups and two coins (named 1 and 2) both of which have a white side and red side. She can shuffle a coin using a dice cup thereby leaving which side comes up top concealed, and she can lift a cup and observe the top facing color of a concealed coin. In this scenario we take a state to be the top facing color of both coins. Further, her action library consists of the actions that represent shuffling the coins and lifting the cups. Say now that in the initial state the two coins are showing their white side, and that neither are hidden by a cup. If the agent shuffles coin 1, the result is that she'll still know that coin 2 is showing its white side, but not whether it is the white or red side of coin 1 that is up. If she subsequently shuffles coin 2, the result is that both coins are concealed by a cup and she is ignorant about which side is up on either coin. This is illustrated on the left in Figure 1.2, where we use arrows to indicate actions and rectangles to indicate belief states.

Here the agent is able to fully predict what will happen, at least from her *own* perspective, where she will become ignorant about which sides of the coins are up. Of course, and this is a fair point, there *is* some actual configuration of the coins, but in planning we take on the role of a non-omniscient agent and model things from this perspective.

When working with belief states, we need to rephrase the notion of applicability. Given a belief state, an action is applicable if its precondition is satisfied in *every* state in the belief state. This is motivated by the sentiment, that if the precondition of an action is not satisfied in some state, then the agent has no way of predicting what happens, hence this is disallowed. Arguably, we can interpret this as modicum of rational agency, because it reflects the disinterest of the agent to act in a manner that makes her knowledge inconsistent.

Example 1.3. We continue the cups and coins scenario where we left off, and now consider the case of lifting the dice cup concealing coin 1. At plan-time the

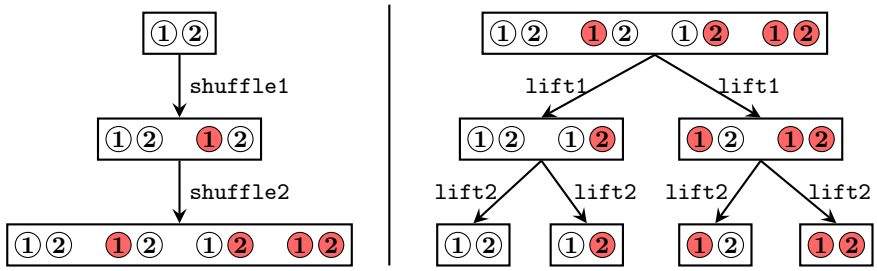


Figure 1.2: An illustration of the coin scenario.

agent can predict that she will come to observe whether the white or red side of the coin is up, but that is all. Nonetheless, she can continue her line of reasoning from both possibilities, and conclude that if she subsequently lifts the other cup, then the exact configuration of the two coins will be known. This is illustrated on the right in Figure 1.2. That actions can lead to different possibilities is indicated by letting a state have two outgoing arrows with identical label.

Abstractly, we can consider a planning problem to represent a succinct labelled state-transition system T, in which the states of T correspond to (belief) states in a planning problem, and transitions are induced by the set of actions and labelled accordingly. Figure 1.2 represents parts of the transition system induced by the cups and coins scenario. When we consider partial observability, states of T are belief states, meaning that T contains a subset of 2^{2^P}; i.e. the collection of all belief states on P. When dealing with nondeterminism where the execution of an action in a state leads to multiple states, then T contains a transition for each possible outcome, for instance as in the case for `lift1` in Example 1.3.[2] This brings us to another important fact about automated planning. The input that is dealt with is that of a planning problem *not* its induced transition system. If we want to provide a genuine alternative to the formalisms that are used in automated planning, bringing *only* a more involved transition system to the table is a nonstarter.

We have not mentioned what constitutes a solution as we made our move away from classical planning, the reason being that a host of solution concepts exist. For instance, a *weak solution* is one in which the course of action *may* lead to a goal, and the more robust *strong solution* is one that *guarantees* that the goal is reached. The third concept we mention is that of a *strong cyclic solution*, which

[2] This classification may seem terminologically off since, after all, it is the shuffle in the dice cup that (if anything) constitutes the nondeterministic part of this problem. We'll quietly accept this anomaly and continue our exposition, though in Chapter 3 we opt to conform with the nomenclature of [Rintanen, 2005].

is guaranteed to *eventually* lead to the goal. In Chapter 6 we propose a solution concept based on qualitative plausibilities, which positions itself somewhere in between weak and strong solutions.

With this we hope to have left the reader with some semblance of what automated planning is all about. Next we complement what we presented above by describing briefly the empiric basis of automated planning, which is a side to automated planning that shouldn't be overlooked.

1.1.1 Automated Planning in Practice

There are many implementations of planning systems in existence, and competitions are held in tandem with the *International Conference on Automated Planning and Scheduling (ICAPS)*, where these systems can be empirically evaluated on a wide range of planning problems. Problems are not formulated as presented here, but rather using the *Planning Domain Definition Language (PDDL)* originating with [Ghallab et al., 1998]. PDDL is a lifted representation of planning problems, based on a typed first-order language with a finite number of predicates and constants, and without functions. By grounding this language we can obtain a planning problem in the set-theoretic formulation given above. The full details are beyond our scope, but the following example serves to illustrate the connection.

Example 1.4. In Figure 1.3 we show how the sliding puzzle problem in Example 1.1 can be specified using PDDL. A planning problem is formulated from a domain specification and a problem specification. The domain specification sets the rules for the planning problem, and the problem instance determines where to start from and where to go.

We start out with the domain specification at the top of Figure 1.3. Here the `:constants` represent locations and pieces, and each constant is assigned a type, which can be seen as an implicit unary predicate. In `:predicates` we specify the predicates for giving meaning to constants. Writing `?x` indicates a free variable that is eligible for substitution with a constant. We use `At` to indicate the location of a piece, `Clear` to specify the location which is unoccupied, and `Neighbor` for encoding the geometry of the puzzle. Before discussing actions, we turn to the the problem specification.

At the bottom of Figure 1.3 is the problem specification. Here `:init` specifies the initial state of the planning problem by listing a number of ground predicates such as `(At 8 l11)`. Taking each ground predicate to be a symbol in P (as used in the set-theoretic formulation), we can see `:init` as specifying the symbols

that are true in the initial state. In fact, a naive translation of this specification would take P to contain one symbol for each possible way in which we can ground a predicate, that is, replace a free variable with a constant. In this case we have $8 \cdot 9 + 9 + 9 \cdot 9 = 162$ symbols after taking types into account (to represent negative literals in preconditions and goals, we actually need to double this number — we omit the details here). The other part of the problem specification is `:goal`, stating the ground predicates (or, symbols) that a goal state must satisfy.

The last part of the domain specification is `:action Move`, which signifies an *action schema*. Similar to how we transformed ground predicates into symbols, action schemas are transformed into actions in the set-theoretic sense by grounding free variables. The naive translation would in this case give rise to $8 \cdot 9 \cdot 9 = 648$ actions (again, additional actions are required as we here have negative literals in preconditions). We forego the details and end our description of the PDDL specification of sliding puzzle problem here.

The above example shows that the input a planning system receives is a high-level description whose translation into the set-theoretic representation incurs significant overhead. On these grounds one might reject the set-theoretic representation as irrelevant (or the other way round), but that would be a hazardous conclusion to draw. In fact, the very popular *Fast Downward Planning System* [Helmert, 2006], upon which many competitive planners are implemented, performs such a translation (albeit in a much more clever manner). A separate point about the use of a high-level specification language is that it reveals structural properties of planning problems, which allows for the extraction of *domain-independent* heuristics. Such heuristics are key if we're interested in constructing generic solvers that scale with the size of the problem instance.

There are two points we'd like to make as we wrap up our exposition. The first is that automated planning deals most prominently with the interplay between states and actions, and this allows us to model the behaviour of a goal-directed (and rational) agent. The second point is that the incorporation of partial observability and nondeterminism into automated planning has been done on the basis of POMDPs, and consequently planning problems have taken on the role as compact representations of POMDPs. The result is that the notions of observability and nondetermism are completely detached, though some form of interdependency can be achieved as mentioned. In Chapter 2 and Chapter 3 we investigate the planning framework that results from using dynamic epistemic logic to capture partial observability and nondeterminism without compromising the basic principles of automated planning.

```
(define (domain SP-3-3)
  (:requirements :strips)
  (:types LOCATION PIECE)
  (:constants l11 l12 l13 l21 l22 l23 l31 l32 l33 - LOCATION 1 2 3 4 5 6 7 8
                                                                  - PIECE )
  (:predicates (At ?x - PIECE ?y - LOCATION ) (Clear ?x - LOCATION)
      (Neighbor ?x ?y - LOCATION) )
  (:action Move
   :parameters (?p - PIECE ?from ?to - LOCATION)
   :precondition (and (At ?p ?from) (Clear ?to) (Neighbor ?from ?to))
   :effect (and
      (not (At ?p ?from) ) (not (Clear ?to) ) (At ?p ?to) (Clear ?from)
   )
  ))



(define (problem SP-A)
(:domain SP-3-3)
(:init  (Neighbor l11 l12) (Neighbor l11 l21)
   (Neighbor l12 l11) (Neighbor l12 l13) (Neighbor l12 l22)
   (Neighbor l13 l12) (Neighbor l13 l23)
   (Neighbor l21 l11) (Neighbor l21 l22) (Neighbor l21 l31)
   (Neighbor l22 l12) (Neighbor l22 l21) (Neighbor l22 l23) (Neighbor l22 l32)
   (Neighbor l23 l13) (Neighbor l23 l22) (Neighbor l23 l33)
   (Neighbor l31 l21) (Neighbor l31 l32)
   (Neighbor l32 l31) (Neighbor l32 l22) (Neighbor l32 l33)
   (Neighbor l33 l32) (Neighbor l33 l23)
   (At 8 l11) (Clear l12) (At 4 l13)
   (At 1 l21) (At 7 l22)  (At 3 l23)
   (At 6 l31) (At 5 l32)  (At 2 l33))
(:goal (and (At 1 l11) (At 2 l12) (At 3 l13) (At 4 l21)
       (At 5 l22) (At 6 l23) (At 7 l31) (At 8 l32)  ) ) )
```

Figure 1.3: PDDL specification of the sliding puzzle problem.

1.2 Dynamic Epistemic Logic

The study of knowledge dates all the way back to antiquity, and has been a major branch of philosophy ever since. Epistemic logic is a proposal for formally dealing with knowledge, and was given its first book-length treatment in [Hintikka, 1962]. To have knowledge of something means for this to be true in every possible world, and to be uncertain means that this "something" is true in some worlds and false in others. In an epistemic language we can talk about knowledge (using an epistemic formula) of simple propositional facts such as *agent a knows that p is true*, but we can also form sentences for talking about higher-order (or meta-) knowledge e.g. *agent a knows that agent b doesn't know whether p is true*. Representing higher-order knowledge is endearing, for many

of our every-day inferences are related to the knowledge of others (we discuss the notion of a theory of mind in Section 1.5).

The notion of knowledge can be formally given with regards to a relational structure, consisting of a set of possible worlds along with a set of accessibility relations (one for each agent) on this set of possible worlds. We add to this a valuation, assigning to each world the basic propositional symbols (such as p) that are true, and we call the result an epistemic model. Statements about knowledge, such as those given above, are then interpreted on epistemic models, with basic facts such as p being determined from the valuation in a world, and (higher-order) knowledge being determined from the interplay between the accessibility relations and the valuation. As epistemic models are formally presented already in Chapter 2 we refrain from providing further details at this point.

Example 1.5. We now return to the coin scenario, and discuss how the belief states in automated planning are related to single-agent epistemic models. We can think of each state in a belief state as representing a possible world. If we further assume that the accessibility relation of an epistemic model is universal, then *what is known in a belief state* coincides with the interpretation of knowledge as truth in every possible world. Belief states therefore capture a certain, idealized, version of knowledge in the single-agent case. Taking on this view, each rectangle illustrated in Figure 1.2 represents a single-agent epistemic model, and so this establishes the first simple connection between automated planning and epistemic logic.

What is missing is how to capture the actions in automated planning is a dynamic component. Fortunately, the idea of extending epistemic logic with dynamic aspects has been investigated to a great extent over the past several decades. In [Fagin et al., 1995] (the culmination of a series of publications by the respective authors) it is shown how the analysis of complex multi-agent systems benefits from ascribing knowledge to agents, with an emphasis on its use in computer science. That epistemic logic, based on questions posed in ancient times, would come into the limelight of computer science demonstrates just how all-pervasive the notion of knowledge is.

One extension of epistemic logic that deals with dynamics of information is [Baltag et al., 1998], often referred to as the "BMS" framework. What is so appealing about this framework is that it can represent a plethora of diverse dynamics, such as obtaining hard evidence, being misinformed, or secretly coming to know something other agents do not. As such the framework is inherently multi-agent, though for the most part we consider the single-agent version. The presentation we give in Chapter 2 is due to [van Ditmarsch and Kooi, 2008] which allows for factual (or ontic) change. We'll refer to this as *dynamic epistemic logic (DEL)*,

even though several logics not based on the BMS framework adhere to this designation. The dynamic component of DEL is introduced via event models, sometimes called actions models or update models. The dynamics is by virtue of the product update operation, in which event models are "multiplied" with epistemic models by taking a restricted Cartesian product of the two structures to form a new epistemic model. On the basis of Example 1.5 we can see event models as a method for transforming one belief state into another — exactly the purpose of actions in automated planning as discussed above. This relationship between event models and actions of automated planning was pointed out in [Löwe et al., 2011, Bolander and Andersen, 2011].

We can assure the reader that we will be dealing both with DEL and related topics in the ensuing chapters, and so we close our discussion after a digression concerning modal logic. The purpose of modal logic (or, modal languages) is to talk about relational structures, namely by its use of Kripke semantics. Kripke semantics coincides with truth in every possible world which we use to ascribe knowledge to an agent, but it can also be used to describe modalities of belief (a weaker form of knowledge), time or morality [Blackburn et al., 2001]. Epistemic logic is therefore a subfield of modal logic, and there is a cross-fertilization of ideas between the two areas, and in many cases results from modal logic carry over directly to epistemic logic. The first point is evident from the investigation of dynamic modal logic in the 1980s which later played a role in development of DEL. The second point is evident as results from modal model theory, for instance those concerning the notion of bisimulation, also plays a fundamental role in the model theory of epistemic models [van Ditmarsch et al., 2007].

1.3 Epistemic Planning

One of the points in Section 1.1 is that automated planning problems are given as an initial state, a number of actions and a goal. In our investigation of planning we take this to be our starting point. What we're after are approaches that extend the foundation of automated planning, in particular by making use of richer logics; e.g. modal logics. Epistemic planning is one such approach and here the richer logic is DEL, which was seminally, and independently, taken in [Löwe et al., 2011, Bolander and Andersen, 2011]. Later works include [Andersen et al., 2012] (the extended version of which constitutes Chapter 2), as well as [Yu et al., 2013, Aucher and Bolander, 2013].

In epistemic planning the initial state is an epistemic model, actions are a number of event models and a goal is an epistemic formula. In [Bolander and Andersen, 2011] the solution to a problem is to find a sequence of event models which,

using the product update operation, produces an epistemic model satisfying the goal formula. The authors show this problem to be decidable for single-agent DEL, but undecidable for multi-agent DEL. In [Löwe et al., 2011] some tractable cases of multi-agent epistemic planning are provided. While their approach does not consider factual change, their notion of a solution does coincide with [Bolander and Andersen, 2011] to the extent that both consider plans as sequences of event models. Recently [Aucher and Bolander, 2013] showed that multi-agent epistemic planning is also undecidable without factual change, while [Yu et al., 2013] identified a fragment of multi-agent epistemic planning that *is* decidable.

Automated planning deals with a single agent environment (depending on whether we interpret nature as an agent), and in this sense each of the aforementioned approaches appear to be immediate generalizations. Here we must make certain reservations. First, with the exception of [Bolander and Andersen, 2011] none of these approaches cater to applicability as described previously. As such, the modicum of rational agency we interpreted from an agent's interest in having consistent knowledge is absent in these presentations. The interpretation in [Yu et al., 2013] is in this sense more appropriate, since it phrases the problem as *explanatory diagnosis*, meaning that the sequence of event models produced represent an explanation of how events may have unfolded, when the "knowledge state" of the system satisfies some epistemic formula (called an observation). We also treat this point in Chapter 2 as the question of an internal versus external perspective. Second, while the formalisms deal with the knowledge of multiple agents, individual autonomous behaviour is not considered. Obviously, automated planning does not provide a solution to these multi-agent aspects, but we do consider it to embody autonomous behaviour of a single agent, and so believe multi-agent planning should deal with each agent exhibiting autonomous behaviour. Third, as solutions are sequences of plans, formalizing notions such as a weak solution or a strong solution are difficult, because they require the agent to make choices as the outcome of actions unfold; e.g. on the basis of what is observed as the coin is lifted in Example 1.3.[3] Chapter 2 extends single-agent epistemic planning with a plan language to tackle this third point.

We point out two references that, in spite of their endearing titles, do not conform to what we take to constitute epistemic planning. The first is "DEL-sequents for regression and epistemic planning" [Aucher, 2012]. Slightly simplified, the problem of epistemic planning is here formulated as finding a single event model (with certain constraints) that produces, from a given epistemic model (initial state), an epistemic model satisfying an epistemic formula (goal). Missing here is that this single event model, while built from atomic events, does not necessarily represent any single action (or sequence of actions) of the

[3] [Bolander and Andersen, 2011] proposes to internalize branching in the event models, implying an extensive modification of the action library. Capturing both weak and strong solution concepts using this approach appears difficult.

action library. The second is "Tractable Multiagent Planning for Epistemic Goals" [van der Hoek and Wooldridge, 2002]. Here the problem that is solved is not finding a solution when given an initial state, an action library and a goal; rather it is finding a solution when given the underlying transition system of such a planning problem. While the promises of tractability are intriguing, it is important to note that this is in terms of the size of the transition system. Furthermore, the transition system is assumed finite which we generally cannot assume in our reading of epistemic planning. In both cases the problems treated are shown to be decidable, which is not compatible with the results of [Bolander and Andersen, 2011].

1.4 Related Work

At the end of each chapter, we discuss work related to the topics that are covered. Our aim now is to provide a broader perspective on things by discussing related approaches for dealing with actions, knowledge, autonomous behaviour and rational agency.

One profound investigation of multi-agent planning is [Brenner and Nebel, 2009], which situates multiple autonomous agents in a dynamic environment. In this setting agents have incomplete knowledge and restricted perceptive capabilities, and to this end each agent is ascribed a belief state. These belief states are based on multi-valued state-variables, which allows for a more sophisticated version of belief state than that discussed in Section 1.1. In a slightly ad-hoc manner, agents can reason about the belief states of other agents. For instance inferring that another agent has come to know the value of a state-variable due to an observation, or by forming mutual belief in the value of a state-variable using speech acts. The authors apply their formalism experimentally, modelling a number of agents operating autonomously in a grid world scenario. We would like to see what could be modelled in this framework when introducing epistemic models as the foundation of each agent's belief state.

We already mentioned the approach of [van der Hoek and Wooldridge, 2002], which more generally pertains to the planning as model checking paradigm. Other cognates are [Cimatti et al., 2003, Bertoli et al., 2001], where symbolic model checking techniques are employed, which allows for compactly representing the state-transition system. Indeed, in some cases the representation of a belief state as a propositional formula is exponentially more succinct. Moreover, it is possible to compute the effects of multiple actions in a single step. As such, this line of research, also used for much richer logics, may prove key to the implementation of efficient planners for planning with more expressive logics.

On the topic of expressive logics, we bring up formalisms that deal with the behaviour of agents over time. Two such logical systems are epistemic temporal logic (ETL) [Parikh and Ramanujam, 2003] and the interpreted systems (IS) of [Fagin et al., 1995], which in [Pacuit, 2007] are shown to be modally equivalent for a language with knowledge, next and until modalities. Under scrutiny in ETL/IS are structures that represent interactive situations involving multi-agents, and may be regarded as a tool for the analysis of social situations, albeit similar to how computation is described using machine code [Pacuit, 2007]. ETL uses the notion of local and global histories. A global history represents a sequence of events that may take place, and local histories result from projecting global histories to the vantage point of individual agents, subject to what events agents are aware of.

[van Benthem et al., 2007] investigates a connection between ETL and DEL, interpreting a sequence of public announcements (a special case of event models) as the temporal evolution of a given initial epistemic model. These sequences of public announcements are constrained so that they conform to a given protocol. The main result is the characterization of the ETL models that are generated by some protocol. Imagining protocols that allow for a more general form of event model and that further capture applicability, we would to a great extent be within the framing of epistemic planning, and one step nearer to mapping out a concrete relationship between the two approaches.

There are even richer logics for reasoning about agents and their group capabilities, such as Coalition Logic [Pauly, 2002], ATL [Alur et al., 2002] and ATEL [van der Hoek and Wooldridge, 2002]. An extensive comparison of all three is [Goranko and Jamroga, 2004], which also presents methods for transforming between the various types of models and languages involved, and further notes the close relationship between lock-step synchronous alternating transition systems and interpreted systems. A characterization of these structures in terms of (sequences of) event models is an interesting yet, to our knowledge, unprobed area. Related is also the concept of "Seeing-to-it-that" (STIT) [Belnap et al., 2001], emanating as a formal theory for clarifying questions in the philosophy of action and the philosophy of norms. Over the past decade most work on STIT theory has been within the field of computer science, for instance by providing results on embedding fragments of STIT logic into formalisms akin to those mentioned in this paragraph [Bentzen, 2010, Broersen et al., 2006a]. A non-exhautive list is [Schwarzentruber and Herzig, 2008, Herzig and Lorini, 2010, Broersen et al., 2006b]. As pointed out in [Benthem and Pacuit, 2013], research into logics of agency has seen a shift from being a "Battle of the Sects" to an emphasis on similarities and relative advantages of the various paradigms. Getting epistemic planning aboard this train might be a pathway towards a multi-agent planning framework dealing with both individual and collective agency.

The work in [Herzig et al., 2003] is close to single-agent epistemic planning. Their complex knowledge states are given syntactically, and represent a set of S5 Kripke models. They introduce ontic actions and epistemic actions as separate notions. The ontic action language is assumed to be given in the situation calculus tradition, so that actions must specify what becomes true at time $t + 1$ based on what is true at time t. An epistemic action is given as a set of possible outcomes; each possible outcome is a propositional formula that represents what the agent comes to know should the particular observation occur. Progression of actions (or, the projection of possible outcomes), and subsequently plans, is given by syntactic manipulation of complex knowledge states. As recently shown in [Lang and Zanuttini, 2013], this approach allows for an exponentially more compact representation of plans; the cost comes at execution time where the evaluation of branching formulas is not constant-time. We recently showed how progression of both epistemic actions and ontic actions can be expressed as event models [Jensen, 2013b], providing insight into the relationship between this approach and single-agent epistemic planning. There seems to be no perspicuous method for extending progression in this framework to a multi-agent setting. Because of this we find DEL and its product update operation to be much more enticing.

An extension of the approach mentioned in the previous paragraph is [Laverny and Lang, 2004] and [Laverny and Lang, 2005]. Here the purely epistemic approach is extended with graded beliefs, that is, belief in some formula to a certain degree. Belief states (as they are, in this context, confusingly called) are introduced via the ordinal conditional functions (or, kappa functions) due to [Spohn, 1988]. What this allows for is modeling scenarios in which an agent reasons about its current beliefs, but more importantly, its future beliefs subject to the actions it can execute. This is similar to our work in Chapter 6, though there are some conceptual differences, with one being (again) that the action language used has no immediate multi-agent counterpart. Another is that in their approach, plans dealing with the same contingency many times over (e.g. a plan that replaces a light bulb, and, in case the replacement is broken, replaces it again — cf. Example 6.1) have a higher chance of success than plans that do not, something which our approach does not naturally handle. There are other non-trivial connections between our approach and that of Laverny and Lang, and further investigation seems prudent.[4]

We end this section by mentioning the "Planning with Knowledge and Sensing planning system" (PKS) due to [Petrick and Bacchus, 2002, Petrick and Bacchus, 2004]. PKS is an implementation of a single-agent planning system using the approach of explicitly modeling the agent's knowledge. The argument is, as

[4] I thank Jérôme Lang for bringing these frameworks to my attention, as well as making a diligent comparison to the work in Chapter 6.

is also stressed in [Bolander and Andersen, 2011], that this is a more natural and general approach to partial observability than variants using e.g. observation variables. An agent's knowledge is compromised of four databases, each of which can be translated to a collection of formulas in epistemic logic (with some first-order extensions), and actions are modelled by modifying these databases separately. They concede that certain configurations of epistemic models cannot be captured, one effect being that their planning algorithm is incomplete. Their argument is pragmatic and PKS was capable of solving then contemporary planning problems. What PKS represents is the bottom-up approach to epistemic planning, taking a fragment of epistemic logic as the basis for extending classical planning to handle partial observability.

1.5 Motivation and Application

The goal of the *Joint Action for Multimodal Embodied Social Systems (JAMES)* project is to develop an *artificial embodied agent that supports socially appropriate, multi-party, multimodal interaction* [JAMES, 2014]. As such it combines many different disciplines belonging to AI, including planning with incomplete information. Concretely, the project considers a robot bartender (agent) that must be able to interact socially intelligent with patrons. Actions available include asking for a drink order and serving it, which requires the generation of plans for obtaining information. This is done in an extension of the PKS framework we described above, which replans when knowledge is not obtained (e.g. the patron mumbles when ordering) or when knowledge is obtained earlier than expected (eg. the patron orders prior to being asked). The social component is in the ability to understand phrases such as "I'll have the same as that guy" or "I'll have another one", which requires a model of each patron. Moreover, the robot bartender must be able to comprehend social norms, for instance when a person asks for five drinks and four bystanders are ignoring it, then it should place the drinks in front of each individual.[5]

While the robot bartender gets by with considering each patron separately, not all social situations are as lenient. We find evidence for this within cognitive science, more specifically in the concept of a *theory of mind* [Premack and Woodruff, 1978]. Theory of mind refers to a mechanism which underlies a crucial aspect of social skills, namely that of being able to conceive of mental states, including what other people know, want, feel or believe. [Baron-Cohen et al., 1985] subjected 61 children to a false-belief task, where each child observed a pretend play involving two doll protagonists Sally and Anne. The scenario is that Sally places a marble in a basket, leaves the scene, after which Anne tranfers

[5]These remarks are based on personal communication with Ron Petrick.

the marble to a box. As Sally returns to the scene, the child is asked where Sally will look for her marble (a belief question), and additionally where the marble really is (a reality question). The children were divided into three groups, namely as either being clinically normal, affected by Down's syndrome or autistic (as determined by a previous test). The passing rate for the belief question (whose correct answer is the basket) for both clinically normal and Down's syndrome children was respectively 85% and 86%, whereas the *failure* rate for the autistic group was 80%. At the same time, every single child correctly answered the reality question.

While the use of false-belief tasks to establish whether a subject has a theory of mind has been subject to criticism [Bloom and German, 2000], this does not entail that the theory of mind mechanism is absent in humans. What it does suggest is that there is more to having a theory of mind than solving false-belief tasks. As epistemic logics, and more generally modal logics, deal with exactly the concepts a human must demonstrate in order to have a theory of mind, it is certainly conceivable that the formalization of this mechanism could be done within these frameworks. The perspectives are both a very general model for studying human and social interaction, as well as artificial agents that are able to function naturally in social contexts.

Admittedly, we're quite a ways off from reaching this point of formalization of social situations, and it may even be utopian to think we'll ever get there. Nonetheless, even small progress in this area may lead to a better formalization of the intricate behaviour of humans. In turn, this allows for enhancing systems that are to interact with humans, including those that are already in commercial existence, which motivates our research from the perspective of application.

1.6 Outline of Thesis

We've now set the tone for for what this thesis deals with. Our first objective is to investigate conditional epistemic planning, where we remove the distinct Markovian mark imprinted on automated planning under partial observability and nondeterminism, replacing it with notions from dynamic epistemic logic. This is the theme in Chapter 2 and Chapter 3. We then turn to the model theoretic aspects of epistemic-plausibility models, and additionally provide results on the relative expressive power of various doxastic languages. This work is conducted in Chapter 4 and Chapter 5, and the results we provide are essential to formulating planning based on beliefs. In Chapter 6 we present a framework that extends the framework of conditional epistemic planning to additionally model doxastic components. This allows us to not only express doxastic goals,

but additionally leads to solution concepts where an agent is only required to find a plan for the outcomes of actions it expects at run-time. The overarching theme is the incorporation of logics for knowledge and belief in planning. The long term perspective of our work is the extension of these formalisms to the multi-agent case, and we end this thesis in Chapter 7 with a brief conclusion and some methodological considerations.

We now summarize the results provided in each chapter of this thesis.

- Chapter 2 is the extended version of [Andersen et al., 2012], appearing here with a few minor corrections. Here we introduce single-agent conditional epistemic planning, based on epistemic models as states, event models as actions and epistemic formulas as goals. We show how a language of conditional plans can be translated into DEL, which entails that plan verification of both weak and strong solutions can be achieved via model checking. Additionally, we present terminating, sound and complete algorithms the synthesis of both weak and strong solutions.

- Chapter 3 delves further into the area of single-agent epistemic planning. We focus on the decision problem of whether a strong solution exists for a planning problem. To this end we consider four types of single-agent epistemic planning problem, separated by the level of observability and whether actions are branching (nondeterministic). We present sound and complete decision procedures for answering the solution existence problem associated with each type. Following this, we show how to construct single-agent epistemic planning problems that simulate Turing machines of various resource bounds. The results show that the solution existence problem associated with the four types of planning problems range in complexity from PSPACE to 2EXP. This coincides with results from automated planning, showing that we're no worse off (asymptotically) using the framework of conditional epistemic planning.

- Chapter 4 is a replicate of [Andersen et al., 2013] (again with a few minor corrections), which plunges deeper into modal territory by looking at single-agent epistemic-plausibility models, that is, epistemic models with the addition of a plausibility relation. Here we take conditional belief as the primitive modality. This leads us to formalize bisimulation for single-agent epistemic-plausibility models in terms of what we name the normal plausibility relation. We show that this notion of bisimulation corresponds with modal equivalence in the typical sense. We then define the semantics of safe belief in terms of conditional belief, more precisely, something is safely believed if the agent continues to believe it no matter what true information it is given. This definition distinguishes itself from other proposals of safe belief in that it is conditional to modally definable subsets,

rather than arbitrary subsets. Additionally, we present a quantitative doxastic modality based on degrees of belief, whose semantics is given with respect to the aforementioned normal plausibility relation.

- Chapter 5 presents additional technical results concerning the framework in Chapter 4. In the first part we show that the notion of bisimulation is also appropriate for the language that extends the epistemic language with only the quantitative doxastic modality, that is, the notion of bisimulation and modal equivalence corresponds on the class of image-finite models. Consequently, modal equivalence for the language with the conditional belief is equivalent to modal equivalence for the language with degrees of belief. In the second part we show that in spite of this, these two languages are expressively incomparable over an infinite set of propositional symbols. What is more, we also show that addition of safe belief to the language with conditional belief yields a more expressive language. The results in this second part (specifically Section 5.2) represent joint work between Mikkel Birkegaard Andersen, Thomas Bolander, Hans van Ditmarsch and this author. This work was conducted during a research visit in Nancy, October 2013, and constitutes part of a joint journal publication not yet finalized. The presentation of these results in this thesis is solely the work of this author.

- Chapter 6 is a replicate of [Andersen et al., 2014]. It adds to the work of Chapter 2 by extending the framework to deal with beliefs and plausibilites. This extension is done on the basis of the logic of doxastic actions [Baltag and Smets, 2008], which extends DEL with a doxastic component. We use this to define the notion of a strong plausibility solution and a weak plausibility solution. These capture solution concepts where an agent needs only to plan for the most plausible outcomes of her actions (as they appear to her at plan-time). Colloquially stated, she doesn't need to plan for the unexpected. As in Chapter 2 we present terminating, sound and complete algorithms for synthesizing solutions.

Chapters 2, 4 and 6 are publications which are coauthored by the author of this thesis. Chapter 3 and Chapter 5 (with the exception of Section 5.2; see above) represents work conducted solely by this author. In Chapter 3 and Chapter 5, we explicitly attribute any definition, construction or proposition to the source from which it is taken and/or adapted. When no such statement is made, the work is due to the author of this thesis.

Chapter 2

Conditional Epistemic Planning

This chapter is the extended version of [Andersen et al., 2012] which appears in the proceedings of the 13th European Conference on Logics in Artificial Intelligence (JELIA), 2012, in Toulouse, France. In the following presentation a few minor corrections have been made.

Conditional Epistemic Planning

Mikkel Birkegaard Andersen Thomas Bolander
Martin Holm Jensen
DTU Compute, Technical University of Denmark

Abstract

Recent work has shown that Dynamic Epistemic Logic (DEL) offers a solid foundation for automated planning under partial observability and non-determinism. Under such circumstances, a plan must branch if it is to guarantee achieving the goal under all contingencies (strong planning). Without branching, plans can offer only the possibility of achieving the goal (weak planning). We show how to formulate planning in uncertain domains using DEL and give a language of conditional plans. Translating this language to standard DEL gives verification of both strong and weak plans via model checking. In addition to plan verification, we provide a tableau-inspired algorithm for synthesising plans, and show this algorithm to be terminating, sound and complete.

2.1 Introduction

Whenever an agent deliberates about the future with the purpose of achieving a goal, she is engaging in the act of planning. When planning, the agent has a view of the environment and knowledge of how her actions affect the environment. Automated Planning is a widely studied area of AI, in which problems are expressed along these lines. Many different variants of planning, with different assumptions and restrictions, have been studied. In this paper we consider

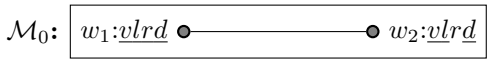


Figure 2.1: The initial situation. The thief is uncertain about whether r holds.

planning under uncertainty (nondeterminism and partial observability), where exact states of affairs and outcomes of actions need not be known by the agent. We formulate such scenarios in an epistemic setting, where states, actions and goals are infused with the notions of knowledge from Dynamic Epistemic Logic (DEL). Throughout this exposition, our running example, starting with Example 2.1, follows the schemings of a thief wanting to steal a precious diamond.

Example 2.1. After following carefully laid plans, a thief has almost made it to her target: The vault containing the invaluable Pink Panther diamond. Standing outside the vault ($\neg v$), she now deliberates on how to get her hands on the diamond (d). She knows the light inside the vault is off ($\neg l$), and that the Pink Panther is on either the right (r) or left ($\neg r$) pedestal inside. Obviously, the diamond cannot be on both the right *and* left pedestal, but nonetheless the agent may be uncertain about its location. This scenario is represented by the epistemic model in Figure 2.1. The edge between w_1 and w_2 signifies that these worlds are indistinguishable to the agent. For visual clarity we omit reflexive edges (each world is always reachable from itself). We indicate with a string the valuation at world w, where an underlined proposition p signifies that p does *not* hold at w.

The agent's goal is to obtain the jewel and to be outside the vault. She can enter and leave the vault, flick the light switch and snatch the contents of either the right or left pedestal. Her aim is to come up with a, possibly conditional, plan, such that she achieves her goal.

By applying DEL to scenarios such as the above, we can construct a procedure for the line of reasoning that is of interest to the thief. In the following section we recap the version of DEL relevant to our purposes. Section 2.3 formalises notions from planning in DEL, allowing verification of plans (using model checking) as either weak or strong solutions. In Section 2.4 we introduce an algorithm for plan synthesis (i.e. generation of plans). Further we show that the algorithm is terminating, sound and complete.

2.2 Dynamic Epistemic Logic

Dynamic epistemic logics describe knowledge and how actions change it. These changes may be epistemic (changing knowledge), ontic (changing facts) or both. The work in this paper deals only with the single-agent setting, though we briefly discuss the multi-agent setting in Section 2.5. As in Example 2.1, agent knowledge is captured by epistemic models. Changes are encoded using event models (defined below). The following concise summary of DEL is meant as a reference for the already familiar reader. The unfamiliar reader may consult [van Ditmarsch and Kooi, 2008, Ditmarsch et al., 2007] for a thorough treatment.

Definition 2.2 (Epistemic Language)**.** Let a set of propositional symbols P be given. The language $\mathcal{L}_{\mathrm{DEL}}(P)$ is given by the following BNF:

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid K\phi \mid [\mathcal{E}, e]\,\phi$$

where $p \in P$, $\mathcal{E}$ denotes an *event model* on $\mathcal{L}_{\mathrm{DEL}}(P)$ as (simultaneously) defined below, and $e \in D(\mathcal{E})$. K is the epistemic modality and $[\mathcal{E}, e]$ the dynamic modality. We use the usual abbreviations for the other boolean connectives, as well as for the dual dynamic modality $\langle \mathcal{E}, e\rangle\,\phi := \neg\,[\mathcal{E}, e]\,\neg\phi$. The dual of K is denoted $\tilde{K}$. $K\phi$ reads as "the (planning) agent knows ϕ" and $[\mathcal{E}, e]\,\phi$ as "after all possible executions of $(\mathcal{E}, e)$, ϕ holds".

Definition 2.3 (Epistemic Models)**.** An *epistemic model* on $\mathcal{L}_{\mathrm{DEL}}(P)$ is a tuple $\mathcal{M} = (W, \sim, V)$, where W is a set of *worlds*, $\sim$ is an equivalence relation (the *epistemic relation*) on W, and $V : P \to 2^W$ is a *valuation*. $D(\mathcal{M}) = W$ denotes the *domain* of $\mathcal{M}$. For $w \in W$ we name $(\mathcal{M}, w)$ a *pointed epistemic model*, and refer to w as the *actual world* of $(\mathcal{M}, w)$.

To reason about the dynamics of a changing system, we make use of *event models*. The formulation of event models we use in this paper is due to van Ditmarsch and Kooi [van Ditmarsch and Kooi, 2008]. It adds ontic change to the original formulation of [Baltag et al., 1998] by adding postconditions to events.

Definition 2.4 (Event Models)**.** An *event model* on $\mathcal{L}_{\mathrm{DEL}}(P)$ is a tuple $\mathcal{E} = (E, \sim, pre, post)$, where

- E is a set of *(basic) events*,
- $\sim \subseteq E \times E$ is an equivalence relation called the *epistemic relation*,
- $pre : E \to \mathcal{L}_{\mathrm{DEL}}(P)$ assigns to each event a *precondition*,
- $post : E \to (P \to \mathcal{L}_{\mathrm{DEL}}(P))$ assigns to each event a *postcondition*.

$D(\mathcal{E}) = E$ denotes the *domain* of $\mathcal{E}$. For $e \in E$ we name $(\mathcal{E}, e)$ a *pointed event model*, and refer to e as the *actual event* of $(\mathcal{E}, e)$.

Definition 2.5 (Product Update)**.** Let $\mathcal{M} = (W, \sim, V)$ and $\mathcal{E} = (E, \sim', pre, post)$ be an epistemic model resp. event model on $\mathcal{L}_{\text{DEL}}(P)$. The *product update* of $\mathcal{M}$ with $\mathcal{E}$ is the epistemic model denoted $\mathcal{M} \otimes \mathcal{E} = (W', \sim'', V')$, where

- $W' = \{(w, e) \in W \times E \mid \mathcal{M}, w \models pre(e)\}$,
- $\sim'' = \{((w, e), (v, f)) \in W' \times W' \mid w \sim v \text{ and } e \sim' f\}$,
- $V'(p) = \{(w, e) \in W' \mid \mathcal{M}, w \models post(e)(p)\}$ for each $p \in P$.

Definition 2.6 (Satisfaction Relation)**.** Let a pointed epistemic model $(\mathcal{M}, w)$ on $\mathcal{L}_{\text{DEL}}(P)$ be given. The satisfaction relation is given by the usual semantics, where we only recall the definition of the dynamic modality:

$$\mathcal{M}, w \models [\mathcal{E}, e]\,\phi \qquad \text{iff } \mathcal{M}, w \models pre(e) \text{ implies } \mathcal{M} \otimes \mathcal{E}, (w, e) \models \phi$$

where $\phi \in \mathcal{L}_{\text{DEL}}(P)$ and $(\mathcal{E}, e)$ is a pointed event model. We write $\mathcal{M} \models \phi$ to mean $\mathcal{M}, w \models \phi$ for all $w \in D(\mathcal{M})$. Satisfaction of the dynamic modality for non-pointed event models $\mathcal{E}$ is introduced by abbreviation, viz. $[\mathcal{E}]\,\phi := \bigwedge_{e \in \mathcal{D}(\mathcal{E})} [\mathcal{E}, e]\,\phi$. Furthermore, $\langle \mathcal{E} \rangle\,\phi := \neg\,[\mathcal{E}]\,\neg\phi$.[1]

Throughout the rest of this paper, all languages (sets of propositional symbols) and all models (sets of possible worlds) considered are implicitly assumed to be finite.

2.3 Conditional Plans in DEL

One way to sum up automated planning is that it deals with the *reasoning side of acting* [Ghallab et al., 2004]. When planning under uncertainty, actions can be nondeterministic and the states of affairs partially observable. In the following, we present a formalism expressing planning under uncertainty in DEL, while staying true to the notions of automated planning. We consider a system similar to that of [Ghallab et al., 2004, sect. 17.4], which motivates the following exposition. The type of planning detailed here is *offline*, where planning is done before acting. All reasoning must therefore be based on the agent's initial knowledge.

[1] Hence, $\mathcal{M}, w \models \langle \mathcal{E} \rangle\,\phi \Leftrightarrow \mathcal{M}, w \models \neg\,[\mathcal{E}]\,\neg\phi \Leftrightarrow \mathcal{M}, w \models \neg(\bigwedge_{e \in \mathcal{D}(\mathcal{E})} [\mathcal{E}, e]\,\neg\phi) \Leftrightarrow \mathcal{M}, w \models \bigvee_{e \in \mathcal{D}(\mathcal{E})} \neg\,[\mathcal{E}, e]\,\neg\phi \Leftrightarrow \mathcal{M}, w \models \bigvee_{e \in D(\mathcal{E})} \langle \mathcal{E}, e \rangle\,\phi$.

Figure 2.2: A model consisting of two information cells

2.3.1 States and Actions: The Internal Perspective

Automated planning is concerned with achieving a certain goal state from a given initial state through some combination of available actions. In our case, states are epistemic models. These models represent situations from the perspective of the planning agent. We call this the *internal perspective*—the modeller is modelling itself. The internal perspective is discussed thoroughly in [Aucher, 2010, Bolander and Andersen, 2011].

Generally, an agent using epistemic models to model its own knowledge and ignorance, will not be able to point out the actual world. Consider the epistemic model $\mathcal{M}_0$ in Figure 2.1, containing two indistinguishable worlds w_1 and w_2. Regarding this model to be the planning agent's own representation of the initial state of affairs, the agent is of course not able to point out the actual world. It is thus natural to represent this situation as a non-pointed epistemic model. In general, when the planning agent wants to model a future (imagined) state of affairs, she does so by a non-pointed model.

The equivalence classes (wrt. $\sim$) of a non-pointed epistemic model are called the *information cells* of that model (in line with the corresponding concept in [Baltag and Smets, 2008]). We generally identify any equivalence class $[w]_\sim$ of a model $\mathcal{M}$ with the submodel it induces, that is, we identify $[w]_\sim$ with $\mathcal{M} \restriction [w]_\sim$. We also use the expression *information cell* on $\mathcal{L}_{\mathrm{DEL}}(P)$ to denote any connected epistemic model on $\mathcal{L}_{\mathrm{DEL}}(P)$, that is, any epistemic model consisting of a single information cell. All worlds in an information cell satisfy the same K-formulas (formulas of the form $K\phi$), thus representing the same situation as seen from the agent's internal perspective. Each information cell of a (non-pointed) epistemic model represents a possible state of knowledge of the agent.

Example 2.7. Recall that our jewel thief is at the planning stage, with her initial information cell $\mathcal{M}_0$. She realises that entering the vault and turning on the light will reveal the location of the Pink Panther. Before actually performing these actions, she can rightly reason that they will lead her to know the location of the diamond, though whether that location is left or right cannot be determined (yet).

Her representation of the possible outcomes of going into the vault and turning on the light is the model $\mathcal{M}'$ in Figure 2.2. The information cells $\mathcal{M}' \restriction \{u_1\}$

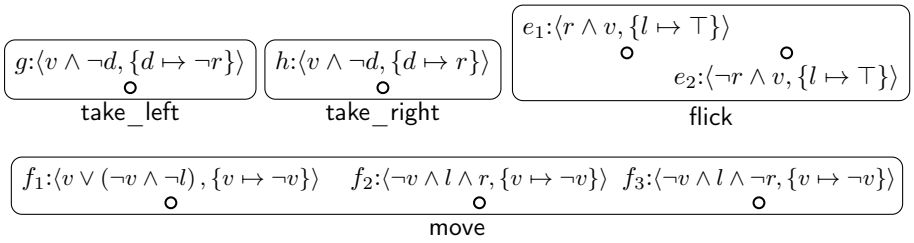


Figure 2.3: Event models representing the actions of the thief

and $\mathcal{M}' \restriction \{u_2\}$ of $\mathcal{M}'$ are exactly the two distinguishable states of knowledge the jewel thief considers possible prior turning the light on in the vault.

In the DEL framework, actions are naturally represented as event models. Due to the internal perspective, these are also taken to be non-pointed. For instance, in a coin toss action, the agent cannot beforehand point out which side will land face up.

Example 2.8. Continuing Example 2.7 we now formalize the actions available to our thieving agent as the event models in Figure 2.3. We use the same conventions for edges as we did for epistemic models. For a basic event e we label it $\langle pre(e), post(e)\rangle$.[2]

The agent is endowed with four actions: take_left, resp. take_right, represent trying to take the diamond from the left, resp. right, pedestal; the diamond is obtained only if it is on the chosen pedestal. Both actions require the agent to be inside the vault and not holding the diamond. flick requires the agent to be inside the vault and turns the light on. Further, it reveals which pedestal the diamond is on. move represents the agent moving in or out of the vault, revealing the location of the diamond provided the light is on.

It can be seen that the epistemic model $\mathcal{M}'$ in Example 2.7 is the result of two successive product updates, namely $\mathcal{M}_0 \otimes \mathsf{move} \otimes \mathsf{flick}$.

2.3.2 Applicability, Plans and Solutions

Reasoning about actions from the initial state as in Example 2.8 is exactly what planning is all about. We have however omitted an important component

[2]For a proposition p whose truth value does not change in e we assume the identity mapping $post(e)(p) = p$, as is also the convention in automated planning.

in the reasoning process, one which is crucial. The notion of *applicability* in automated planning dictates when the outcomes of an action are defined. The idea translates to DEL by insisting that no world the planning agent considers possible is eliminated by the product update of an epistemic model with an event model.

Definition 2.9 (Applicability)**.** An event model $\mathcal{E}$ is said to be *applicable* in an epistemic model $\mathcal{M}$ if $\mathcal{M} \models \langle \mathcal{E} \rangle \top$.

This concept of applicability is easily shown to be equivalent with the one defined in [Bolander and Andersen, 2011] when restricting the latter to the single-agent case. However, for our purposes of describing plans as formulas, we need to express applicability as formulas as well. The discussion in [de Lima, 2007, sect. 6.6] also notes this aspect, insisting that actions must be meaningful. The same sentiment is expressed by our notion of applicability.

The situation in Example 2.7 calls for a way to express conditional plans. Clearly, our agent can only snatch the jewel from the correct pedestal conditioned on how events unfold when she acts. To this end we introduce a language for conditional plans allowing us to handle such contingencies.

Definition 2.10 (Plan Language)**.** Given a finite set A of event models on $\mathcal{L}_{\mathrm{DEL}}(P)$, the *plan language* $\mathcal{L}_{\mathrm{P}}(P, \mathsf{A})$ is given by:

$$\pi ::= \mathcal{E} \mid \mathsf{skip} \mid \mathsf{if}\ K\phi\ \mathsf{then}\ \pi\ \mathsf{else}\ \pi \mid \pi; \pi$$

where $\mathcal{E} \in \mathsf{A}$ and $\phi \in \mathcal{L}_{\mathrm{DEL}}(P)$. We name members π of this language *plans*, and use $\mathsf{if}\ K\phi\ \mathsf{then}\ \pi$ as shorthand for $\mathsf{if}\ K\phi\ \mathsf{then}\ \pi\ \mathsf{else}\ \mathsf{skip}$.

The reading of the plan constructs are "do $\mathcal{E}$", "do nothing", "if $K\phi$ then π, else π'", and "first π then π'" respectively. Note that the condition of the if-then-else construct is required to be a K-formula. This is to ensure that the planning agent can only make her choices of actions depend on worlds that are distinguishable to her (cf. the discussion of the internal perspective in Section 2.3.1). The idea is similar to the *meaningful plans* of [de Lima, 2007], where branching is only allowed on *epistemically interpretable formulas*.

An alternative way of specifying conditional plans is *policies*, where (in our terminology) each information cell maps to an event model [Ghallab et al., 2004, Sect. 16.2]. There are slight differences between the expressiveness of conditional plans and policies (e.g. policies can finitely represent repetitions); our main motivation for not using policies is that it would require an enumeration of each information cell of the planning domain.

Definition 2.11 (Translation)**.** We define a *strong translation* $[\![\cdot]\!]_s \cdot$ and a *weak translation* $[\![\cdot]\!]_w \cdot$ as functions from $\mathcal{L}_{\mathrm{P}}(P, \mathsf{A}) \times \mathcal{L}_{\mathrm{DEL}}(P)$ into $\mathcal{L}_{\mathrm{DEL}}(P)$ by:

$$
\begin{aligned}
&[\![\mathcal{E}]\!]_s \, \phi := \langle \mathcal{E} \rangle \top \wedge [\mathcal{E}] \, K\phi \\
&[\![\mathcal{E}]\!]_w \, \phi := \langle \mathcal{E} \rangle \top \wedge \tilde{K} \, \langle \mathcal{E} \rangle \, K\phi \\
&[\![\mathsf{skip}]\!]. \, \phi := \phi \\
&[\![\mathsf{if}\ \phi' \ \mathsf{then}\ \pi \ \mathsf{else}\ \pi']\!]. \, \phi := (\phi' \to [\![\pi]\!]. \, \phi) \wedge (\neg\phi' \to [\![\pi']\!]. \, \phi) \\
&[\![\pi; \pi']\!]. \, \phi := [\![\pi]\!]. \, ([\![\pi']\!]. \, \phi)
\end{aligned}
$$

Plans describe the manner in which actions are carried out. We interpret plans π relative to a formula ϕ and want to answer the question of whether or not π achieves ϕ. Using Definition 2.11 we can answer this question by verifying truth of the DEL formula provided by the translations. This is supported by the results of Section 2.4. We concisely read $[\![\pi]\!]_s \, \phi$ as "π achieves ϕ", and $[\![\pi]\!]_w \, \phi$ as "π may achieve ϕ" (elaborated below). By not specifying separate semantics for plans our framework is kept as simple as possible. Note that applicability (Definition 2.9) is built into the translations through the occurrence of the conjunct $\langle \mathcal{E} \rangle \top$ in both the strong translation $[\![\mathcal{E}]\!]_s \, \phi$ and the weak translation $[\![\mathcal{E}]\!]_w \, \phi$.

The difference between the two translations relate to the *robustness* of plans: $[\![\pi]\!]_s \, \phi$, resp. $[\![\pi]\!]_w \, \phi$, means that every step of π is applicable and that following π always leads, resp. may lead, to a situation where ϕ is known.

Definition 2.12 (Planning Problems and Solutions)**.** Let P be a finite set of propositional symbols. A planning problem on P is a triple $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ where

- $\mathcal{M}_0$ is an information cell on $\mathcal{L}_{\mathrm{DEL}}(P)$ called the *initial state*.
- A is a finite set of event models on $\mathcal{L}_{\mathrm{DEL}}(P)$ called the *action library*.
- $\phi_g \in \mathcal{L}_{\mathrm{DEL}}(P)$ is the *goal (formula)*.

We say that a plan $\pi \in \mathcal{L}_{\mathrm{P}}(P, \mathsf{A})$ is a *strong solution* to $\mathcal{P}$ if $\mathcal{M}_0 \models [\![\pi]\!]_s \, \phi_g$, a *weak solution* if $\mathcal{M}_0 \models [\![\pi]\!]_w \, \phi_g$ and not a solution otherwise.

Planning problems are defined with the sentiment we've propagated in our examples up until now. The agent is presently in $\mathcal{M}_0$ and wishes ϕ_g to be the case. To this end, she reasons about the actions (event models) in her action library A, creating a conditional plan. Using model checking, she can verify whether this plan is either a weak or strong solution, since plans translate into formulas of $\mathcal{L}_{\mathrm{DEL}}(P)$. Further, [van Ditmarsch and Kooi, 2008] gives reduction axioms for DEL-formulas, showing that any formula containing the dynamic modality can be expressed as a formula in (basic) epistemic logic. Consequently, plan verification can be seen simply as epistemic reasoning about $\mathcal{M}_0$.

Example 2.13. We continue our running example by discussing it formally as a planning problem and considering the solutions it allows. The initial state is still $\mathcal{M}_0$, and the action library $\mathsf{A} = \{\mathsf{flick}, \mathsf{move}, \mathsf{take\_left}, \mathsf{take\_right}\}$. We discuss the plans below and their merit for our thief.

- $\pi_1 = \mathsf{flick}; \mathsf{move}; \mathsf{if}\ Kr\ \mathsf{then}\ \mathsf{take\_right}\ \mathsf{else}\ \mathsf{take\_left}; \mathsf{move}$
- $\pi_2 = \mathsf{move}; \mathsf{take\_right}; \mathsf{move}$
- $\pi_3 = \mathsf{move}; \mathsf{flick}; \mathsf{take\_right}; \mathsf{move}$
- $\pi_4 = \mathsf{move}; \mathsf{flick}; \mathsf{if}\ Kr\ \mathsf{then}\ \mathsf{take\_right}\ \mathsf{else}\ \mathsf{take\_left}; \mathsf{move}$

We consider two planning problems varying only on the goal formula, $\mathcal{P}_1 = (\mathcal{M}_0, \mathsf{A}, d \wedge \neg v)$ and $\mathcal{P}_2 = (\mathcal{M}_0, \mathsf{A}, \tilde{K}d \wedge \neg v)$. In $\mathcal{P}_1$ her goal is to obtain the diamond and be outside the vault, whereas in $\mathcal{P}_2$ she wishes to be outside the vault *possibly* having obtained the diamond.

Let $\pi_1' = \mathsf{move}; \mathsf{if}\ Kr\ \mathsf{then}\ \mathsf{take\_right}\ \mathsf{else}\ \mathsf{take\_left}; \mathsf{move}$ and note that $\pi_1 = \mathsf{flick}; \pi_1'$. Using the strong translation of π_1, we get $\mathcal{M}_0 \models [\![\pi_1]\!]_s \phi_g$ iff $\mathcal{M}_0 \models \langle\mathsf{flick}\rangle \top \wedge [\mathsf{flick}] [\![\pi_1']\!]_s \phi_g$. As $\mathcal{M}_0 \models \langle\mathsf{flick}\rangle \top$ does not hold, π_1 is not a solution. This is expected, since flicking the switch in the initial state is not an applicable action. Verifying that π_2 is a strong solution to $\mathcal{P}_2$ amounts to checking if $\mathcal{M}_0 \models [\![\pi_2]\!]_s \tilde{K}d \wedge \neg v$ which translates to

$$\mathcal{M}_0 \models \langle\mathsf{move}\rangle \top \wedge [\mathsf{move}]\, K \left(\langle\mathsf{take\_right}\rangle \top \wedge [\mathsf{take\_right}]\, K \left(\langle\mathsf{move}\rangle \top \wedge [\mathsf{move}]\, K \left(\tilde{K}d \wedge \neg v \right) \right) \right)$$

With the same approach we can conclude that π_2 is not a solution to $\mathcal{P}_1$, π_3 is a weak solution to $\mathcal{P}_1$ and $\mathcal{P}_2$, and π_4 is a strong solution to $\mathcal{P}_1$ and $\mathcal{P}_2$.

2.4 Plan Synthesis

We now show how to synthesise conditional plans for solving planning problems. To synthesise plans, we need a mechanism for coming up with formulas characterising information cells for if-then-else constructs to branch on. Inspired by [Barwise and Moss, 1996, van Benthem, 1998], these are developed in the following. Proofs are omitted, as they are straightforward and similar to proofs in the aforementioned references.

Definition 2.14 (Characterising Formulas)**.** Let $\mathcal{M} = (W, \sim, V)$ denote an information cell on $\mathcal{L}_{\mathrm{DEL}}(P)$. We define for all $w \in W$ a formula ϕ_w by: $\phi_w = \bigwedge_{p \in V(w)} p \wedge \bigwedge_{p \in P - V(w)} \neg p$. We define the *characterising formula* for $\mathcal{M}$, $\delta_{\mathcal{M}}$, as follows: $\delta_{\mathcal{M}} = K(\bigwedge_{w \in W} \tilde{K}\phi_w \wedge K \bigvee_{w \in W} \phi_w)$.

Lemma 2.15. *Let $\mathcal{M}$ be an information cell on $\mathcal{L}_{\text{DEL}}(P)$. Then for all epistemic models $\mathcal{M}' = (W', \sim', V')$ and all $w' \in W'$ we have that $(\mathcal{M}', w') \models \delta_{\mathcal{M}}$ if and only if there exists a $w \in \mathcal{D}(\mathcal{M})$ such that $(\mathcal{M}, w) \leftrightarroweq (\mathcal{M}', w')$.*[3]

For purposes of synthesis, we use the product update solely on non-pointed epistemic and event models. Lemma 2.16 shows that satisfaction of the dynamic modality for non-pointed event models in non-pointed epistemic models relates to the product update in the obvious way.

Lemma 2.16. *Let $\mathcal{M}$ be an epistemic model and $\mathcal{E}$ an event model. Then $\mathcal{M} \models [\mathcal{E}]\phi$ iff $\mathcal{M} \otimes \mathcal{E} \models \phi$.*

Proof. $\mathcal{M} \models [\mathcal{E}]\phi \;\Leftrightarrow\;$ for all $w \in \mathcal{D}(\mathcal{M}) : \mathcal{M}, w \models [\mathcal{E}]\phi \Leftrightarrow$
for all $w \in \mathcal{D}(\mathcal{M}) : \mathcal{M}, w \models \bigwedge_{e \in \mathcal{D}(\mathcal{E})} [\mathcal{E}, e]\phi \Leftrightarrow$
for all $(w, e) \in \mathcal{D}(\mathcal{M}) \times \mathcal{D}(\mathcal{E}) : \mathcal{M}, w \models [\mathcal{E}, e]\phi \Leftrightarrow$
for all $(w, e) \in \mathcal{D}(\mathcal{M}) \times \mathcal{D}(\mathcal{E}) : \mathcal{M}, w \models pre(e)$ implies $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \phi \Leftrightarrow$
for all $(w, e) \in D(\mathcal{M} \otimes \mathcal{E}) : \mathcal{M} \otimes \mathcal{E}, (w, e) \models \phi \;\Leftrightarrow\; \mathcal{M} \otimes \mathcal{E} \models \phi$. □

2.4.1 Planning Trees

When synthesising plans, we explicitly construct the search space of the problem as a labelled AND-OR tree, a familiar model for planning under uncertainty [Ghallab et al., 2004]. Our AND-OR trees are called *planning trees*.

Definition 2.17. A *planning tree* is a finite, labelled AND-OR tree in which each node n is labelled by an epistemic model $\mathcal{M}(n)$, and each edge (n, m) leaving an OR-node is labelled by an event model $\mathcal{E}(n, m)$.

Planning trees for planning problems $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ are constructed as follows. Let the initial planning tree T_0 consist of just one OR-node $root(T_0)$ with $\mathcal{M}(root(T_0)) = \mathcal{M}_0$ (the root labels the initial state). A planning tree for $\mathcal{P}$ is then any tree that can be constructed from T_0 by repeated applications of the following non-deterministic tree expansion rule.

Definition 2.18 (Tree Expansion Rule)**.** Let T be a planning tree for a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$. The tree expansion rule is defined as follows. Pick an OR-node n in T and an event model $\mathcal{E} \in \mathsf{A}$ applicable in $\mathcal{M}(n)$ with the proviso that $\mathcal{E}$ does not label any existing outgoing edges from n. Then:

[3] Here $(\mathcal{M}, w) \leftrightarroweq (\mathcal{M}', w)$ denotes that $(\mathcal{M}, w)$ and $(\mathcal{M}', w)$ are bisimilar according to the standard notion of bisimulation on pointed epistemic models.

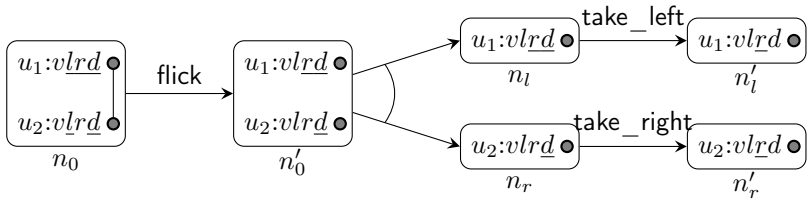


Figure 2.4: Planning tree for a variant of the Pink Panther problem.

1. Add a new node m to T with $\mathcal{M}(m) = \mathcal{M}(n) \otimes \mathcal{E}$, and add an edge (n, m) with $\mathcal{E}(n, m) = \mathcal{E}$.

2. For each information cell $\mathcal{M}'$ in $\mathcal{M}(m)$, add an OR-node m' with $\mathcal{M}(m') = \mathcal{M}'$ and add the edge (m, m').

The tree expansion rule is similar in structure to—and inspired by—the expansion rules used in tableau calculi, e.g. for modal and description logics [Horrocks et al., 2006]. Note that the expansion rule applies only to OR-nodes, and that an applicable event model can only be used once at each node.

Considering single-agent planning a two-player game, a useful analogy for planning trees are game trees. At an OR-node n, the agent gets to pick any applicable action $\mathcal{E}$ it pleases, winning if it ever reaches an epistemic model in which the goal formula holds (see the definition of solved nodes further below). At an AND-node m, the environment responds by picking one of the information cells of $\mathcal{M}(m)$—which of the distinguishable outcomes is realised when performing the action.

Example 2.19. In Fig. 2.4 is a planning tree for a variant of the Pink Panther planning problem, this one where the thief is already inside the vault. The root is n_0. Three applications of the tree expansion rule have been made, the labels on edges indicating the chosen action. n_0, n_l and n_r are OR-nodes. n'_0, n'_l and n'_r are AND-nodes. The child nodes of the latter two AND-nodes have been omitted, as their information cell is the same as that of their parent nodes. Pay particular attention to how flick reveals the location of the diamond. In the initial state, $\mathcal{M}(n_0) \models \neg Kr \wedge \neg K\neg r$, while $\mathcal{M}(n'_0) \models Kr \vee K\neg r$, $\mathcal{M}(n_l) \models K\neg r$ and $\mathcal{M}(n_r) \models Kr$.

Without restrictions on the tree expansion rule, even very simple planning problems might be infinitely expanded. Finiteness of trees (and therefore termination) is ensured by the following blocking condition.

$\mathcal{B}_1$ The tree expansion rule may not be applied to a node n for which there exists an ancestor node m with $\mathcal{M}(m) \leftrightarrows \mathcal{M}(n)$.[4]

A planning tree for a planning problem $\mathcal{P}$ is called $\mathcal{B}_1$-*saturated* if no more expansions are possible satisfying condition $\mathcal{B}_1$.

Lemma 2.20 (Termination)**.** *Any procedure that builds a $\mathcal{B}_1$-saturated planning tree for a planning problem $\mathcal{P}$ by repeated application of the tree expansion rule terminates.*

Proof. Planning trees built by repeated application of the tree expansion rule are finitely branching: the action library is finite, and every epistemic model has only finitely many information cells. Furthermore, condition $\mathcal{B}_1$ ensures that no branch has infinite length: there only exists finitely many mutually non-bisimilar epistemic models over any given finite set of propositional symbols [Bolander and Andersen, 2011]. König's Lemma now implies finiteness of the planning tree.

□

Definition 2.21 (Solved Nodes)**.** Let T be any (not necessarily saturated) planning tree for a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$. By recursive definition, a node n in T is called *solved* if one of the following holds:

- $\mathcal{M}(n) \models \phi_g$ (the node satisfies the goal formula).
- n is an OR-node having at least one solved child.
- n is an AND-node having all its children solved.

Continuing the game tree analogy, we see that a solved node corresponds to one for which there exists a winning strategy. Regardless of the environment's choice, the agent can achieve its goal. Let T and $\mathcal{P}$ be as above. Below we show that when a node n is solved, it is possible to construct a (strong) solution to the planning problem $(\mathcal{M}(n), \mathsf{A}, \phi_g)$. In particular, if the root node is solved, a strong solution to $\mathcal{P}$ can be constructed. As it is never necessary to expand a solved node, nor any of its descendants, we can augment the blocking condition $\mathcal{B}_1$ in the following way.

$\mathcal{B}_2$ The tree expansion rule may not be applied to a node n if one of the following holds: 1) n is solved; 2) n has a solved ancestor; 3) n has an ancestor node m with $\mathcal{M}(m) \leftrightarrows \mathcal{M}(n)$.

[4] Here $\mathcal{M}(m) \leftrightarrows \mathcal{M}(n)$ denotes that $\mathcal{M}(m)$ and $\mathcal{M}(n)$ are bisimilar according to the standard notion of bisimulation between non-pointed epistemic models.

In the following, we will assume that all planning trees have been built according to $\mathcal{B}_2$. One consequence is that a solved OR-node has exactly one solved child. We make use of this in the following definition.

Definition 2.22 (Plans for Solved Nodes)**.** Let T be any planning tree for $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$. For each solved node n in T, a plan $\pi(n)$ is defined recursively by:

- if $\mathcal{M}(n) \models \phi_g$, then $\pi(n) = \mathsf{skip}$.
- if n is an OR-node and m its solved child, then $\pi(n) = \mathcal{E}(n, m); \pi(m)$.
- if n is an AND-node with children $m_1, \ldots, m_k$, then $\pi(n) =$ $\mathsf{if}\ \delta_{\mathcal{M}(m_1)}\ \mathsf{then}\ \pi(m_1)\ \mathsf{else\ if}\ \ \delta_{\mathcal{M}(m_2)}\ \mathsf{then}\ \pi(m_2)\ \mathsf{else}\ \cdots\ \mathsf{if}\ \delta_{\mathcal{M}(m_k)}\ \mathsf{then}\ \pi(m_k)$

Example 2.23. For the goal of achieving the diamond, $\phi_g = d$, we have that the root n_0 of the planning tree of Figure 2.4 is solved, as both n_l' and n_r' satisfy the goal formula. Definition 2.22 gives us

$$\pi(n_0) = \mathsf{flick};\ \mathsf{if}\ \delta_{\mathcal{M}(n_l)}\ \mathsf{then\ take\_left; skip\ else\ if}\ \delta_{\mathcal{M}(n_r)}\mathsf{then\ take\_right; skip}$$

This plan can easily be shown to be a strong solution to the planning problem of achieving d from the initial state $\mathcal{M}(n_0)$. In our soundness result below, we show that plans of solved roots are always strong solutions to their corresponding planing problems.

Theorem 2.24 (Soundness)**.** *Let T be a planning tree for a problem $\mathcal{P}$ such that $root(T)$ is solved. Then $\pi(root(T))$ is a strong solution to $\mathcal{P}$.*

Proof. We need to prove that $\pi(root(T))$ is a strong solution to $\mathcal{P}$, that is, $\mathcal{M}_0 \models [\![\pi(root(T))]\!]_s\, \phi_g$. Since $\mathcal{M}_0$ is the label of the root, this can be restated as $\mathcal{M}(root(T)) \models [\![\pi(root(T))]\!]_s\, \phi_g$. To prove this fact, we will prove the following stronger claim:

- For each solved node n in T, $\mathcal{M}(n) \models [\![\pi(n)]\!]_s\, \phi_g$.

We prove this by induction on the height of n. The base case is when n is a leaf. Since n is solved, we must have $\mathcal{M}(n) \models \phi_g$. In this case $\pi(n) = \mathsf{skip}$. From $\mathcal{M}(n) \models \phi_g$ we can conclude $\mathcal{M}(n) \models [\![\mathsf{skip}]\!]_s\, \phi_g$, that is, $\mathcal{M}(n) \models [\![\pi(n)]\!]_s\, \phi_g$. This covers the base case. For the induction step, assume that for all solved nodes m of height $< h$, $\mathcal{M}(m) \models [\![\pi(m)]\!]_s\, \phi_g$. Let n be an arbitrary solved node n of height h. We then need to show $\mathcal{M}(n) \models [\![\pi(n)]\!]_s\, \phi_g$. We have two cases to consider, depending on whether n is an AND- or an OR-node.

Case 1: n is an AND-node. Let $m_1, \ldots, m_k$ be the children of n. By definition, all of these are solved. We have $\pi(n) = \mathsf{if}\ \delta_{\mathcal{M}(m_1)}\ \mathsf{then}\ \pi(m_1)\ \mathsf{else\ if}\ \delta_{\mathcal{M}(m_2)}\ \mathsf{then}$

$\pi(m_2)$ else $\cdots$ if $\delta_{\mathcal{M}(m_k)}$ then $\pi(m_k)$ else skip. The induction hypothesis gives us $\mathcal{M}(m_i) \models [\![\pi(m_i)]\!]_s \phi_g$ for all $i = 1, \ldots, k$.

Claim 1. $\mathcal{M}(n) \models \delta_{\mathcal{M}(m_i)} \to [\![\pi(m_i)]\!]_s \phi_g$, for all $i = 1, \ldots, k$.

Proof of claim. Let $w \in \mathcal{D}(\mathcal{M}(n))$ be chosen arbitrarily. We then need to prove that if $\mathcal{M}(n), w \models \delta_{\mathcal{M}(m_i)}$ then $\mathcal{M}(n), w \models [\![\pi(m_i)]\!]_s \phi_g$. Assuming $\mathcal{M}(n), w \models \delta_{\mathcal{M}(m_i)}$, we get from Lemma 2.15 that there must be a $w' \in \mathcal{D}(\mathcal{M}(m_i))$ such that $\mathcal{M}(m_i), w' \leftrightarrows \mathcal{M}(n), w$. Since $\mathcal{M}(m_i) \models [\![\pi(m_i)]\!]_s \phi_g$, in particular we get $\mathcal{M}(m_i), w' \models [\![\pi(m_i)]\!]_s \phi_g$, and thus $\mathcal{M}(n), w \models [\![\pi(m_i)]\!]_s \phi_g$.

Claim 2. $\mathcal{M}(n) \models \bigvee_{i=1,\ldots,k} \delta_{\mathcal{M}(m_i)}$.

Proof of claim. Let $w \in \mathcal{D}(\mathcal{M}(n))$ be chosen arbitrarily. We then need to prove that $\mathcal{M}(n), w \models \vee_{i=1,\ldots,k} \delta_{\mathcal{M}(m_i)}$. Since $w \in \mathcal{D}(\mathcal{M}(n))$ it must belong to one of the information cells of $\mathcal{M}(n)$, that is, $w \in \mathcal{D}(\mathcal{M}(m_j))$ for some j. Thus $\mathcal{M}(n), w \leftrightarrows \mathcal{M}(m_j), w$. From Lemma 2.15 we then get $\mathcal{M}(n), w \models \delta_{\mathcal{M}(m_j)}$, and thus $\mathcal{M}(n), w \models \vee_{i=1,\ldots,k} \delta_{\mathcal{M}(m_i)}$.

From (1) and (2), we now get:

$$\mathcal{M}(n) \models \bigwedge_{i=1,\ldots,k} (\delta_{\mathcal{M}(m_i)} \to [\![\pi(m_i)]\!]_s \phi_g) \wedge \bigvee_{i=1,\ldots,k} \delta_{\mathcal{M}(m_i)} \Rightarrow$$

$$\mathcal{M}(n) \models \bigwedge_{i=1,\ldots,k} \big(\delta_{\mathcal{M}(m_i)} \wedge \bigwedge_{j=1,\ldots,i-1} \neg\delta_{\mathcal{M}(m_j)} \to [\![\pi(m_i)]\!]_s \phi_g\big) \wedge \big(\bigwedge_{i=1,\ldots,k} \neg\delta_{\mathcal{M}(m_i)} \to [\![\mathsf{skip}]\!]_s \phi_g\big) \Rightarrow$$

$$\begin{aligned}
\mathcal{M}(n) \models\ & (\delta_{\mathcal{M}(m_1)} \to [\![\pi(m_1)]\!]_s \phi_g)\ \wedge (\neg\delta_{\mathcal{M}(m_1)} \to \\
& (\delta_{\mathcal{M}(m_2)} \to [\![\pi(m_2)]\!]_s \phi_g) \wedge (\neg\delta_{\mathcal{M}(m_2)} \to \\
& \quad \cdots \\
& \quad (\delta_{\mathcal{M}(m_k)} \to [\![\pi(m_k)]\!]_s \phi_g) \wedge (\neg\delta_{\mathcal{M}(m_k)} \to \\
& \quad\ [\![\mathsf{skip}]\!]_s \phi_g) \cdots) \Rightarrow
\end{aligned}$$

$$\begin{aligned}
\mathcal{M}(n) \models\ & [\![\mathsf{if}\ \delta_{\mathcal{M}(m_1)}\ \mathsf{then}\ \pi(m_1)\ \mathsf{else} \\
& \quad \mathsf{if}\ \delta_{\mathcal{M}(m_2)}\ \mathsf{then}\ \pi(m_2)\ \mathsf{else} \\
& \quad \cdots \\
& \quad\ \mathsf{if}\ \delta_{\mathcal{M}(m_k)}\ \mathsf{then}\ \pi(m_k)\ \mathsf{else} \\
& \quad\ \mathsf{skip}]\!]\ \phi_g \Rightarrow
\end{aligned}$$

$$\mathcal{M}(n) \models [\![\pi(n)]\!]_s \phi_g.$$

Case 2: n is an OR-node. Here we have $\pi(n) = \mathcal{E}(n, m); \pi(m)$ for the solved child m of n. The induction hypothesis gives $\mathcal{M}(m) \models [\![\pi(m)]\!]_s \phi_g$, and hence $\mathcal{M}(m) \models K\,[\![\pi(m)]\!]_s \phi_g$. We now show $\mathcal{M}(n) \models [\![\pi(n)]\!]_s \phi_g$. Since, by definition, $\mathcal{M}(m) = \mathcal{M}(n) \otimes \mathcal{E}(n, m)$, we get $\mathcal{M}(n) \otimes \mathcal{E}(n, m) \models K\,[\![\pi(m)]\!]_s \phi_g$. We can now

apply Lemma 2.16 to conclude $\mathcal{M}(n) \models [\mathcal{E}(n,m)]\, K\, [\![\pi(m)]\!]_s\, \phi_g$. By definition, $\mathcal{E}(n,m)$ must be applicable in $\mathcal{M}(n)$, that is, $\mathcal{M}(n) \models \langle \mathcal{E}(n,m) \rangle \top$. Thus we now have $\mathcal{M}(n) \models \langle \mathcal{E}(n,m) \rangle \top \wedge [\mathcal{E}(n,m)]\, K\, [\![\pi(m)]\!]_s\, \phi_g$. Using Definition 2.11, we can rewrite this as $\mathcal{M}(n) \models [\![\mathcal{E}(n,m)]\!]_s\, [\![\pi(m)]\!]_s\, \phi_g$. Using Definition 2.11 again, we get $\mathcal{M}(n) \models [\![\mathcal{E}(n,m); \pi(m)]\!]_s\, \phi_g$, and thus finally $\mathcal{M}(n) \models [\![\pi(n)]\!]_s\, \phi_g$, as required. □

Theorem 2.25 (Completeness)**.** *If there is a strong solution to the planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$, then a planning tree T for $\mathcal{P}$ can be constructed, such that root(T) is solved.*

Proof. We first prove the following claim.

Claim 1. If (if ϕ then π_1 else π_2) is a strong solution to $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$, then so is π_1 or π_2.

Proof of claim. Assume (if ϕ then π_1 else π_2) is a strong solution to $(\mathcal{M}_0, \mathsf{A}, \phi_g)$, that is, $\mathcal{M}_0 \models [\![\text{if } \phi \text{ then } \pi_1 \text{ else } \pi_2]\!]_s\, \phi_g$. Then, by definition, $\mathcal{M}_0 \models (\phi \rightarrow [\![\pi_1]\!]_s\, \phi_g) \wedge (\neg\phi \rightarrow [\![\pi_2]\!]_s\, \phi_g)$. Since $\mathcal{M}_0$ is an information cell, and ϕ is a K-formula, we must have either $\mathcal{M}_0 \models \phi$ or $\mathcal{M}_0 \models \neg\phi$. Thus we get that either $\mathcal{M}_0 \models [\![\pi_1]\!]_s\, \phi_g$ or $\mathcal{M}_0 \models [\![\pi_2]\!]_s\, \phi_g$, as required.

Note that we have $[\![\mathsf{skip}; \pi]\!]_s\, \phi_g = [\![\mathsf{skip}]\!]_s\, ([\![\pi]\!]_s\, \phi_g) = [\![\pi]\!]_s\, \phi_g$. Thus, we can without loss of generality assume that no plan contains a subexpression of the form $\mathsf{skip}; \pi$. The length of a plan π, denoted $|\pi|$, is defined recursively by: $|\mathsf{skip}| = 1$; $|\mathcal{E}| = 1$; $|\text{if } \phi \text{ then } \pi_1 \text{ else } \pi_2| = |\pi_1| + |\pi_2|$; $|\pi_1; \pi_2| = |\pi_1| + |\pi_2|$.

Claim 2. Let π be a strong solution to $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ with $|\pi| \geq 2$. Then there exists a strong solution of the form $\mathcal{E}; \pi'$ with $|\mathcal{E}; \pi'| \leq |\pi|$.

Proof of claim. Proof by induction on $|\pi|$. The base case is $|\pi| = 2$. We have two cases, $\pi =$ if ϕ then π_1 else π_2 and $\pi = \pi_1; \pi_2$, both with $|\pi_1| = |\pi_2| = 1$. If π is the latter, it already has desired the form. If $\pi =$ if ϕ then π_1 else π_2 we have by Claim 1 that either π_1 or π_2 is a strong solution to $\mathcal{P}$. Thus also either $\pi_1; \mathsf{skip}$ or $\pi_2; \mathsf{skip}$ is a strong solution to $\mathcal{P}$, and both of these have length $|\pi|$. This completes the base case. For the induction step, we assume that if π', with $|\pi'| < l$, is a strong solution to a planning problem $\mathcal{P}'$, then there exists is a strong solution of the form $(\mathcal{E}; \pi'')$, with $|\mathcal{E}; \pi''| \leq |\pi'|$. Now consider a plan π of length l which is a strong solution to $\mathcal{P}$. We again have two cases to consider, $\pi =$ if ϕ then π_1 else π_2 and $\pi = \pi_1; \pi_2$. If $\pi = \pi_1; \pi_2$ is a strong solution to $\mathcal{P}$, then π_1 is a strong solution to the planning problem $\mathcal{P}' = (\mathcal{M}_0, \mathsf{A}, [\![\pi_2]\!]_s\, \phi_g)$, as $\mathcal{M}_0 \models [\![\pi_1; \pi_2]\!]_s\, \phi_g \Leftrightarrow \mathcal{M}_0 \models [\![\pi_1]\!]_s\, [\![\pi_2]\!]_s\, \phi_g$. Clearly $|\pi_1| < l$, so the induction hypothesis gives that there is a strong solution $(\mathcal{E}; \pi_1')$

to $\mathcal{P}'$, with $|\mathcal{E}; \pi_1'| \leq |\pi_1|$. Then, $\mathcal{E}; \pi_1'; \pi_2$ is a strong solution to $\mathcal{P}$ and we have $|\mathcal{E}; \pi_1'; \pi_2| = |\mathcal{E}; \pi_1'| + |\pi_2| \leq |\pi_1| + |\pi_2| = |\pi|$. If $\pi = \mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ is a strong solution to $\mathcal{P}$, then we have by Claim 1 that either π_1 or π_2 is a strong solution to $\mathcal{P}$. With both $|\pi_1| < l$ and $|\pi_2| < l$, the induction hypothesis gives the existence a strong solution $\mathcal{E}; \pi'$, with $|\mathcal{E}; \pi'| \leq |\pi|$. This completes the proof of the claim.

We now prove the theorem by induction on $|\pi|$, where π is a strong solution to $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$. We need to prove that there exists a planning tree T for $\mathcal{P}$ in which the root is solved. Let T_0 denote the planning tree for $\mathcal{P}$ only consisting of its root node with label $\mathcal{M}_0$. The base case is when $|\pi| = 1$. Here, we have two cases, $\pi = \mathsf{skip}$ and $\pi = \mathcal{E}$. In the first case, the planning tree T_0 already has its root solved, since $\mathcal{M}_0 \models [\![\mathsf{skip}]\!]_s \phi_g \Leftrightarrow \mathcal{M}_0 \models \phi_g$. In the second case $\pi = \mathcal{E}$. Since π is a strong solution to $\mathcal{P}$, we have $\mathcal{M}_0 \models [\![\mathcal{E}]\!]_s \phi_g$, that is, $\mathcal{M}_0 \models \langle \mathcal{E} \rangle \top \wedge [\mathcal{E}] K \phi_g$. Thus $\mathcal{E}$ is applicable in $\mathcal{M}_0$ meaning that we can apply the tree expansion rule to T_0, which will produce an AND-node m with $\mathcal{E}(root(T_0), m) = \mathcal{E}$ and $\mathcal{M}(m) = \mathcal{M}_0 \otimes \mathcal{E}$. Call the expanded tree T_1. Since we have $\mathcal{M}_0 \models [\mathcal{E}] K \phi_g$, Lemma 2.16 gives us $\mathcal{M}_0 \otimes \mathcal{E} \models K \phi_g$, that is, $\mathcal{M}(m) \models K \phi_g$, and hence $\mathcal{M}(m) \models \phi_g$. This implies that $\mathcal{M}(m)$ and thus $root(T_1)$ is solved. The base case is hereby completed.

For the induction step, assume that a planning tree with solved root can be constructed for problems with strong solutions of length $< l$. Let π be a strong solution to $\mathcal{P}$ with $|\pi| = l$. By Claim 2, there exists a strong solution of the form $\mathcal{E}; \pi'$ with $|\mathcal{E}; \pi'| \leq |\pi|$. As $\mathcal{M}_0 \models [\![\mathcal{E}; \pi']\!]_s \phi_g \Leftrightarrow \mathcal{M}_0 \models [\![\mathcal{E}]\!]_s [\![\pi']\!]_s \phi_g \Leftrightarrow \mathcal{M}_0 \models \langle \mathcal{E} \rangle \top \wedge [\mathcal{E}] K([\![\pi']\!]_s \phi_g)$, the tree expansion rule can be applied by picking $\mathcal{E}$ and $\mathcal{M}_0$. This produces the AND-node m with $\mathcal{E}(n, m) = \mathcal{E}$ and $\mathcal{M}(m) = \mathcal{M}_0 \otimes \mathcal{E}$. $m_1, \ldots, m_k$ are the children of m, and $\mathcal{M}(m_i) = \mathcal{M}_i$ the information cells in $\mathcal{M}(m)$. From $\mathcal{M}_0 \models [\mathcal{E}] K([\![\pi']\!]_s \phi_g)$ we get $\mathcal{M}_0 \otimes \mathcal{E} \models K [\![\pi']\!]_s \phi_g$, using Lemma 2.16. This implies $\mathcal{M}_i \models K [\![\pi']\!]_s \phi_g$, and hence $\mathcal{M}_i \models [\![\pi']\!]_s \phi_g$, for each information cell $\mathcal{M}_i$ of $\mathcal{M}(m) = \mathcal{M}_0 \otimes \mathcal{E}$. Thus π' must be a strong solution to each of the planning problems $\mathcal{P}_i = (\mathcal{M}_i, \mathcal{A}, \phi_g)$. As $|\pi'| < |\mathcal{E}; \pi'| \leq l$, the induction hypothesis gives that planning trees T_i with solved roots can be constructed for each $\mathcal{P}_i$. Let T denote T_0 expanded with $m, m_1, \ldots, m_k$, and each T_i be the subtree rooted at m_i. Then each of the nodes m_i are solved in T, and in turn both m and $root(T)$ are solved. □

2.4.2 Strong Planning Algorithm

With all the previous in place, we now have an algorithm for synthesising strong solutions for planning problems $\mathcal{P}$, given as follows.

StrongPlan($\mathcal{P}$)

1 Let T be the plan. tree only consisting of $root(T)$ labelled by the init. state of $\mathcal{P}$.
2 Repeatedly apply the tree expansion rule of $\mathcal{P}$ to T until it is $\mathcal{B}_2$-saturated.
3 If $root(T)$ is solved, return $\pi(root(T))$, otherwise return FAIL.

Theorem 2.26. StrongPlan($\mathcal{P}$) *is a terminating, sound and complete algorithm for producing strong solutions to planning problems. Soundness means that if* StrongPlan($\mathcal{P}$) *returns a plan, it is a strong solution to* $\mathcal{P}$*. Completeness means that if* $\mathcal{P}$ *has a strong solution,* StrongPlan($\mathcal{P}$) *will return one.*

Proof. Termination comes from Lemma 2.20 (with $\mathcal{B}_1$ replaced by the stronger condition $\mathcal{B}_2$), soundness from Theorem 2.24 and completeness from Theorem 2.25 (given any two saturated planning trees T_1 and T_2 for the same planning problem, the root node of T_1 is solved iff the root node of T_2 is). □

2.4.3 Weak Planning Algorithm

With few changes, the machinery already in place gives an algorithm for synthesising weak solutions. Rather than requiring all children of an AND-node be solved, we require only one. This corresponds to the notion of weak, defined in Definition 2.11. Only one possible execution need lead to the goal.

Definition 2.27 (Weakly Solved Nodes)**.** A node n is called *weakly solved* if either $\mathcal{M}(n) \models \phi_g$ or n has at least one weakly solved child.

We keep the tree expansion rule, but make use of a new blocking condition $\mathcal{B}_3$ using Definition 2.27 rather than Definition 2.21.

Definition 2.28 (Plans for Weakly Solved Nodes)**.** Let T be any planning tree for $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$. For each weakly solved node n in T, a plan $\pi_w(n)$ is defined recursively by:

- if $\mathcal{M}(n) \models \phi_g$, then $\pi_w(n) = \mathsf{skip}$
- if n is an OR-node and m its weakly solved child, then $\pi_w(n) = \mathcal{E}(n, m); \pi_w(m)$
- if n is an AND-node and m its weakly solved child, then $\pi_w(n) = \pi_w(m)$

The algorithm for weak planning is defined as follows.

WeakPlan($\mathcal{P}$)

1 Let T be the plan. tree only consisting of $root(T)$ labelled by the init. state of $\mathcal{P}$.
2 Repeatedly apply the tree expansion rule of $\mathcal{P}$ to T until it is $\mathcal{B}_3$-saturated.
3 If $root(T)$ is weakly solved, return $\pi_w(root(T))$, otherwise return FAIL.

Theorem 2.29. WeakPlan($\mathcal{P}$) *is a terminating, sound and complete algorithm for producing weak solutions to planning problems.*

2.5 Related and Future Work

In this paper, we have presented a syntactic characterisation of weak and strong solutions to epistemic planning problems, that is, we have characterised solutions as formulas. [Bolander and Andersen, 2011] takes a semantic approach to strong solutions for epistemic planning problems. In their work plans are sequences of actions, requiring conditional choice of actions at different states to be encoded in the action structure itself. We represent choice explicitly, using a language of conditional plans. An alternative to our approach of translating conditional plans into formulas of DEL would be to translate plans directly into (complex) event models. This is the approach taken in [Baltag and Moss, 2004], where they have a language of epistemic programs similar to our language of plans (modulo the omission of ontic actions). Using this approach in a planning setting, one could translate each possible plan π into the corresponding event model $\mathcal{E}(\pi)$, check its applicability, and check whether $\mathcal{M}_0 \otimes \mathcal{E}(\pi) \models \phi_g$ (the goal is satisfied in the product update of the initial state with the event model). However, even for a finite action library, there are infinitely many distinct plans, and thus infinitely many induced event models to consider when searching for a solution. To construct a terminating planning algorithm with this approach, one would still have to limit the plans considered (e.g. by using characterising formulas), and also develop a more involved loop-checking mechanism working at the level of plans. Furthermore, our approach more obviously generalises to algorithms for replanning, which is current work.

The meaningful plans of [de Lima, 2007, chap. 2] are reminiscent of the work in this paper. Therein, plan verification is cast as validity of an EDL-consequence in a given system description. Like us, they consider single-agent scenarios, conditional plans, applicability and incomplete knowledge in the initial state. Unlike us, they consider only deterministic actions. In the multi-agent treatment [de Lima, 2007, chap. 4], action laws are translated to a fragment of DEL with only public announcements and public assignments, making actions singleton event models. This means foregoing nondeterminism and therefore sensing actions.

Planning problems in [Löwe et al., 2011] are solved by producing a sequence of pointed event models where an external variant of applicability (called *possible at*) is used. Using such a formulation means outcomes of actions are fully determined, making conditional plans and weak solutions superfluous. As noted by the authors, and unlike our framework, their approach does not consider factual change. We stress that [Bolander and Andersen, 2011, Löwe et al., 2011, de Lima, 2007] all consider the multi-agent setting which we have not treated here.

In our work so far, we haven't treated the problem of where domain formulations come from, assuming just that they are given. Standardised description languages are vital if modal logic-based planning is to gain wide acceptance in the planning community. Recent work worth noting in this area includes [Baral et al., 2012], which presents a specification language for the multi-agent belief case.

As suggested by our construction of planning trees, there are several connections between our approach and two-player imperfect information games. First, product updates imply perfect recall [van Benthem, 2001]. Second, when the game is at a node belonging to an information set, the agent knows a proposition only if it holds throughout the information set; corresponding to our use of information cells. Finally, the strong solutions we synthesise are very similar to mixed strategies. A strong solution caters to any information cell (contingency) it may bring about, by selecting exactly one sub-plan for each [Aumann and Hart, 1992].

Our work naturally relates to [Ghallab et al., 2004], where the notions of strong and weak solutions are found. Their belief states are sets of states which may be partioned by observation variables. Our partition of epistemic models into information cells follows straight from the definition of product update. A clear advantage in our approach is that actions encode both nondetermism and partial observability. [Rintanen, 2004] shows that for conditional planning (prompted by nondeterministic actions) in partially observable domains the *plan existence problem* is 2-EXP-complete (plans must succeed with probability 1; i.e. be strong solutions). StrongPlan($\mathcal{P}$) implicitly answers the same question for $\mathcal{P}$ (it gives a strong solution if one exists). Reductions between the two decision problem variants would give a complexity measure of our approach, and also formally link conditional epistemic planning with the approaches used in automated planning.

We would like to do plan verification and synthesis in the multi-agent settings. We believe that generalising the notions introduced in this paper to multi-pointed epistemic and event models are key. Plan synthesis in the multi-agent setting is undecidable [Bolander and Andersen, 2011], but considering

restricted classes of actions as is done in [Löwe et al., 2011] seems a viable route for achieving decidable multi-agent planning. Another interesting area is to consider modalities such as plausibility and preferences. This would allow an agent to plan for (perhaps only) the most likely outcomes of its own actions and the preferred actions taken by other agents in the system. This could then be combined with the possibility of doing replanning, as mentioned above.

Chapter 3

Addendum to Conditional Epistemic Planning

In this chapter we expand on the results of Chapter 2, putting an emphasis on the computational complexity of the following decision problem.

Definition 3.1. Let $\mathcal{P}$ be a planning problem on P. The *solution existence problem (for $\mathcal{P}$)* is the decision problem of whether there exists a strong solution to $\mathcal{P}$.

Given that our focus is solely on strong solutions, we often omit the qualification and simply write solution it its place. As in the previous chapter, we consider only single-agent conditional epistemic planning. The solution existence problem has been widely studied within the field of automated planning, and we feel obliged to already point out, that this chapter will only show that things are as *expected* when we base planning problems on dynamic epistemic logic. What we mean by expected is that the types of problems we consider range from being PSPACE-Complete to 2EXP-Complete, coinciding with results from automated planning [Bylander, 1994, Littman, 1997, Haslum and Jonsson, 1999, Rintanen, 2004]. At the same time, our exposition proves that we're no worse off in terms of computational complexity when using the framework of conditional epistemic planning. Additionally our method provides insights into how planning can be lifted to work with (dynamic) modal logics, which becomes valuable to us in later chapters where we make a slight change of scenery and consider epistemic-doxastic notions.

We lead out with some technical preliminaries in Section 3.1, and then proceed in

Section 3.2 to illustrate that the algorithm introduced in the previous chapter for synthesizing strong plans is suboptimal. As a natural follow-up, we use Section 3.3 to present two alternative procedures that provide better upper bounds. That there is little hope of improving these procedures in terms of asymptotic complexity is shown in Section 3.4, where we construct planning problems for simulating Turing machines of various resource bounds. We conclude upon our findings in Section 3.5, and further discuss the impact of our results.

3.1 Technical Preliminaries

In what follows we introduce notions used throughout the remainder of this chapter, as well as later on in Chapter 5. We first present several known results from the vast literature on dynamic epistemic logic, and add a few simple findings of our own which prove to be useful. We then talk about the planning formalism we have presented and spell out the types of planning problems we will be investigating. We round things off with introducing alternating Turing machines, which will serve as our prime model of computation, as well as the complexity classes relevant to our exposition.

3.1.1 Equivalence

We begin by giving very broad definitions of equivalence between formulas resp. models, as presented in [van Ditmarsch et al., 2007].[1]

Definition 3.2. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages interpretable on the same class of models, and let $\phi_1 \in \mathcal{L}_1$ and $\phi_2 \in \mathcal{L}_2$. We say that ϕ_1 and ϕ_2 are *equivalent*, in symbols $\phi_1 \equiv \phi_2$, if for any pointed model $(\mathcal{M}, w)$ of the given class, we have that $\mathcal{M}, w \models \phi_1$ iff $\mathcal{M}, w \models \phi_2$.

Definition 3.3. Given two models $\mathcal{M}, \mathcal{M}'$ belonging to the same class of models and a language $\mathcal{L}$ interpreted on this class, for any $w \in D(\mathcal{M})$ and $w' \in D(\mathcal{M}')$, we say that $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ are *modally equivalent* for $\mathcal{L}$ if:

$$\mathcal{M}, w \models \phi \;\Leftrightarrow\; \mathcal{M}', w' \models \phi \quad \text{for all } \phi \in \mathcal{L}$$

and denote this $(\mathcal{M}, w) \equiv_{\mathcal{L}} (\mathcal{M}', w')$. Likewise if

$$\mathcal{M} \models \phi \;\Leftrightarrow\; \mathcal{M}' \models \phi \quad \text{for all } \phi \in \mathcal{L}$$

then we write $\mathcal{M} \equiv_{\mathcal{L}} \mathcal{M}'$[2].

[1] The broad nature of Definition 3.3 and 3.2 allow us to reuse them in Chapter 5.

[2] We can overload $\equiv$ without ambiguity since it is clear from context whether we're referring to (pointed) models or formulas.

In the remainder of this chapter we let $\mathcal{L}_{\mathrm{EL}}(P)$ denote the static epistemic sublanguage of $\mathcal{L}_{\mathrm{DEL}}(P)$, that is, without the dynamic modality $[\mathcal{E}, e]\,\phi$. Shown already in the the seminal [Baltag et al., 1998], dynamic epistemic logic can be translated into static epistemic logic, and this is indeed also the case for the dialect of DEL we employ here.

Proposition 3.4. *There exists an equivalence-preserving translation function* $tr : \mathcal{L}_{\mathrm{DEL}}(P) \to \mathcal{L}_{\mathrm{EL}}(P)$ *s.t.* $\phi \equiv tr(\phi)$ *for any* $\phi \in \mathcal{L}_{\mathrm{DEL}}(P)$.

Proof. We take the translation of action models in [van Ditmarsch et al., 2007, Sect. 7.6] which handle event models without postconditions. To accommodate postconditions we need only modify the cases of $[\mathcal{E}, e]\,p$ (proposition) and $[\mathcal{E}, e]\,[\mathcal{E}', e']\,\phi$ (composition), in accordance with the proof system in [van Ditmarsch and Kooi, 2008]. From soundness of the proof system it follows that the translation is equivalence-preserving. □

Since $\mathcal{L}_{\mathrm{EL}}(P)$ is a sublanguage of $\mathcal{L}_{\mathrm{DEL}}(P)$, it follows from Proposition 3.4 that the two languages are equally expressive, intuitively meaning that $\mathcal{L}_{\mathrm{EL}}(P)$ and $\mathcal{L}_{\mathrm{DEL}}(P)$ can distinguish the exact same models (we discuss this notion formally in Chapter 5). Therefore many results from epistemic logic immediately carry over to dynamic epistemic logic, and we proceed to give a few such results.

3.1.2 Bisimulation

Recall that the valuation of an epistemic model assigns a set of worlds to each symbol. An alternative to this is to use a labelling function, assigning instead a set of symbols to each world in the model. The *labelling function* L of an epistemic model $\mathcal{M} = (W, \sim, V)$ on P is given by $L : W \to 2^P$, such that for all $w \in W$, $L(w) = \{p \in P \mid w \in V(p)\}$ is the *label of* w. The two notions are interdefinable and we will work with whichever of the two suits our needs the best. As a case in point we use labelling functions for introducing one of the key tools of the trade to a modal logician, the notion of bisimulation (see for instance [Blackburn et al., 2001]).[3]

Definition 3.5. Let epistemic models $\mathcal{M} = (W, \sim, V)$ and $\mathcal{M}' = (W', \sim', V')$ on P be given with labelling functions L, L'. A non-empty relation $\mathfrak{R} \subseteq W \times W'$ is a *bisimulation* between $\mathcal{M}$ and $\mathcal{M}'$ if for all $(w, w') \in \mathfrak{R}$:

[atoms] $L(w) = L(w')$.

[3] Also, this treatment is more compatible with the presentation in Chapter 4 and 5, where we downright take the labelling function as primitive in our models.

[forth] If $v \in W$ and $w \sim v$, there is a $v' \in W'$ s.t. $w' \sim' v'$ and $(v, v') \in \mathfrak{R}$.

[back] If $v' \in W'$ and $w' \sim' v'$, there is a $v \in W$ s.t. $w \sim v$ and $(v, v') \in \mathfrak{R}$.

If a bisimulation $\mathfrak{R}$ between $\mathcal{M}$ and $\mathcal{M}'$ exists such that $(w, w') \in \mathfrak{R}$, we say that $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ are *bisimilar* and write $(\mathcal{M}, w) \leftrightarroweq (\mathcal{M}', w')$. Similarly, if a bisimulation $\mathfrak{R}$ between $\mathcal{M}$ and $\mathcal{M}'$ exists whose domain is W and codomain is W', we say that $\mathcal{M}$ and $\mathcal{M}'$ are *totally bisimilar* and write $\mathcal{M} \leftrightarroweq \mathcal{M}'$. When there is no risk of confusion we sometimes simply write bisimilar to mean totally bisimilar.

The result below (observe that we consider *finite* models) is known to hold for $\mathcal{L}_{\mathrm{EL}}(P)$ (see for instance [Blackburn et al., 2001, Theorem 2.20, 2.24]). We can readily show the same result for $\mathcal{L}_{\mathrm{DEL}}(P)$, using Proposition 3.4 and the fact that $\mathcal{L}_{\mathrm{EL}}(P)$ is a sublanguage of $\mathcal{L}_{\mathrm{DEL}}(P)$.

Lemma 3.6. *Given finite epistemic models $\mathcal{M}$ and $\mathcal{M}'$ on P, we have that:*

$$\begin{array}{rcl} (\mathcal{M}, w) \leftrightarroweq (\mathcal{M}', w') & \Leftrightarrow & (\mathcal{M}, w) \equiv_{\mathcal{L}_{\mathrm{DEL}}(P)} (\mathcal{M}', w') \\ \mathcal{M} \leftrightarroweq \mathcal{M}' & \Leftrightarrow & \mathcal{M} \equiv_{\mathcal{L}_{\mathrm{DEL}}(P)} \mathcal{M}' \end{array}$$

for any $w \in D(\mathcal{M})$ and $w' \in D(\mathcal{M}')$.

From Definition 2.9 we see that applicability is invariant under bisimulation.

Corollary 3.7. *Given finite epistemic models $\mathcal{M}$ and $\mathcal{M}'$ on P s.t. $\mathcal{M} \leftrightarroweq \mathcal{M}'$, and an event model $\mathcal{E}$ of $\mathcal{L}_{\mathrm{DEL}}(P)$, then $\mathcal{E}$ is applicable in $\mathcal{M}$ iff $\mathcal{E}$ is applicable in $\mathcal{M}'$.*

Bisimulation is also the right fit when we apply the tree expansion rule (as well as the graph expansion rule defined in Section 3.3), that is, bisimilarity is preserved by the product update operation. The proof is as in [van Ditmarsch et al., 2007, Prop. 6.21], with Lemma 3.6 giving us that preconditions and postconditions assigned to atoms are modally equivalent for any two bisimilar worlds of $\mathcal{M}$ and $\mathcal{M}'$.

Lemma 3.8. *Given finite epistemic models $\mathcal{M}$ and $\mathcal{M}'$ on P and an event model $\mathcal{E}$ of $\mathcal{L}_{\mathrm{DEL}}(P)$. Consider any $w \in D(\mathcal{M})$ and $w' \in D(\mathcal{M}')$ and some $e \in D(\mathcal{E})$ s.t. $\mathcal{M}, w \models pre(e)$. If $(\mathcal{M}, w) \leftrightarroweq (\mathcal{M}', w')$ then $(\mathcal{M} \otimes \mathcal{E}, (w, e)) \leftrightarroweq (\mathcal{M}' \otimes \mathcal{E}, (w', e))$. Further, if there is a total bisimulation between $\mathcal{M}$ and $\mathcal{M}'$, and $\mathcal{E}$ is applicable in $\mathcal{M}$, then there is a total bisimulation between $\mathcal{M} \otimes \mathcal{E}$ and $\mathcal{M}' \otimes \mathcal{E}$.*

In this rather simple logic we have a straightforward characterization of bisimilarity.

Lemma 3.9. *Given an epistemic model* $\mathcal{M} = (W, \sim, V)$ *and* $w, w' \in W$*, if* $w \sim w'$ *then we have that* $(\mathcal{M}, w) \leftrightarroweq (\mathcal{M}, w')$ *iff* $L(w) = L(w')$*.*

Proof. Assuming $(\mathcal{M}, w) \leftrightarroweq (\mathcal{M}, w')$ we can immediately use [atoms] and see that $L(w) = L(w')$. For the other direction assume that $L(w) = L(w')$ and let $\mathfrak{R} = \{(v, v) \mid v \in [w]_\sim\} \cup \{(w, w')\}$, that is, $\mathfrak{R}$ is the identity relation on the $\sim$-equivalence class of w with the addition of (w, w'). We show that $\mathfrak{R}$ is a bisimulation between $(\mathcal{M}, w)$ and $(\mathcal{M}, w')$. By Definition 3.5, each pair $(v, v) \in \mathfrak{R}$ relating a world to itself immediately satisfies the three conditions of an (auto-)bisimulation, hence we need only consider $(w, w') \in \mathfrak{R}$. From assumption of $L(w) = L(w')$ we have that (w, w') satisfies [atoms]. To show [forth] we consider any $u \in W$ s.t. $w \sim u$. This means that $u \in [w]_\sim$ and we have that:

- If $u = w$ then by symmetry of $\sim$, we have $w' \sim w$ and by construction $(w, w') \in \mathfrak{R}$,
- Otherwise, from $w \sim w'$ we have that $[w]_\sim = [w']_\sim$, and so $w \sim u$ implies that $w' \sim u$ and by construction we have $(u, u) \in \mathfrak{R}$.

[back] is shown symmetrically. □

We'll be interested in models that are not bisimilar to any *smaller* model (the minimal representation of a model). Obtaining such a model in the general case is done via the so-called *bisimulation-contraction*. To this end we can consider the *maximal* bisimulation between a model and itself (a maximal auto-bisimulation), and then take the quotient of a model wrt. to this auto-bisimulation [Blackburn et al., 2006]. Following [Ågotnes et al., 2012] we call such models *contraction-minimal*. Given a contraction-minimal model $\mathcal{M}$ we have that for any two distinct worlds w, w' of $\mathcal{M}$, $(\mathcal{M}, w)$ and $(\mathcal{M}, w')$ are *not* bisimilar. Turning now to some simple combinatorics, we assume that P contains n propositional symbols for the following three results.

Lemma 3.10. *The number of non-totally bisimilar singleton epistemic models on* P *is* 2^n*.*

Proof. A singleton model contains exactly one world, hence the notion of bisimilarity and total bisimilarity coincides. By Lemma 3.9 we therefore have that two singleton models are not totally bisimilar iff the labels of their respective

worlds are different. A label is a subset of P, hence the number of singleton epistemic models which are non-totally bisimilar is 2^n (the size of the power set of P). □

Lemma 3.11. *There are at most 2^n worlds in a contraction-minimal information cell on P.*

Proof. No two worlds are bisimilar in any contraction-minimal information cell $\mathcal{M}$, and so from $\sim$ being universal and Lemma 3.9 it follows that no two worlds in $\mathcal{M}$ have the same label. As there are 2^n different labels this also bounds the number of worlds in a contraction-minimal information cell. □

Lemma 3.12. *The number of non-totally bisimilar information cells on P is $2^{(2^n)}$.*

Proof. We can without loss of generality count the number of contraction-minimal information cells. We can identify each contraction-minimal information cell with the set of labels assigned to worlds. With 2^n different labels we therefore have $2^{(2^n)}$ different sets of labels, hence giving the number of information cells on P that are not totally bisimilar. □

3.1.3 Size of Inputs

To properly analyze the computational complexity it is necessary to clearly state the size of the inputs given to a particular problem. In our case this means the size of formulas, epistemic models and event models. We postulate that the following quantities are all "reasonable" as per [Garey and Johnson, 1979]. The first quantity reflects that we can encode epistemic models as labelled graphs.

Definition 3.13. Given an epistemic model $\mathcal{M} = (W, \sim, V)$ on P, its size $|\mathcal{M}|$ is given by $|W| + |\sim| + |W| \cdot |P|$.

The following definition is adapted from [Aucher and Schwarzentruber, 2012] to also encompass postconditions.

Definition 3.14. Given a formula $\phi \in \mathcal{L}_{\mathrm{DEL}}(P)$ and an event model $(\mathcal{E}, e)$ of $\mathcal{L}_{\mathrm{EL}}(P)$ with $\mathcal{E} = (E, \sim, pre, post)$. Their sizes $|\phi|$ and $|(\mathcal{E}, e)|$ are given by the

following simultaneous recursion:

$$\begin{aligned}
|p| &= 1 \\
|\neg\phi| &= 1 + |\phi| \\
|\phi \wedge \psi| &= 1 + |\phi| + |\psi| \\
|K\phi| &= 1 + |\phi| \\
|\,[\mathcal{E}, e]\,\phi| &= |\mathcal{E}| + |\phi| \\
|\mathcal{E}| &= |E| + |\sim| + \sum_{e' \in E} (|pre(e')| + (\sum_{p' \in P} |post(e')(p')|))
\end{aligned}$$

3.1.4 Model Checking and the Product Update Operation

We now turn our attention to model checking, which is a task necessary to compute the product update. The input of a model checking problem is a finite epistemic model $\mathcal{M}$, a world $w \in D(\mathcal{M})$ and a formula ϕ interpretable on epistemic models. The local formulation of model checking asks whether $\mathcal{M}, w \models \phi$, and the global formulation whether $\mathcal{M} \models \phi$. Observe that the local formulation is used in the definition of product update, and the global formulation for determining (given an information cell) whether an event model is applicable and whether the goal formula is satisfied. We can answer the global formulation using a procedure for the local formulation, simply by doing local model checking in each world of $\mathcal{M}$. The overhead (linear in the size of $\mathcal{M}$) in computational effort of this generalization is sufficiently small for us to ignore the difference here.

For formulas in $\mathcal{L}_{\mathrm{EL}}(P)$ we have an algorithm for model checking whose running time is polynomial in the size of the input formula and model [Blackburn et al., 2006, Section 1.4.1]. Given a formula in $\phi \in \mathcal{L}_{\mathrm{DEL}}(P)$, we can solve the model checking task by running the algorithm on $tr(\phi)$, that is, the translation from $\mathcal{L}_{\mathrm{DEL}}(P)$ to $\mathcal{L}_{\mathrm{EL}}(P)$. However, as for instance observed by [Aucher and Schwarzentruber, 2012], the size of $tr(\phi)$ might be exponential in the size of ϕ. With this approach the worst case running time of the algorithm is therefore exponential in the size of ϕ.

Proposition 3.15. *Let a pointed epistemic model $(\mathcal{M}, w)$ be given. If $\phi \in \mathcal{L}_{\mathrm{EL}}(P)$ there is an algorithm for deciding $\mathcal{M}, w \models \phi$ whose running time is polynomial in $|\mathcal{M}|$ and $|\phi|$. If $\phi \in \mathcal{L}_{\mathrm{DEL}}(P)$ there is an algorithm for deciding $\mathcal{M}, w \models \phi$ whose running time is polynomial in $|\mathcal{M}|$ and $2^{|\phi|}$.*

Using a slightly different dialect of $\mathcal{L}_{\mathrm{DEL}}(P)$ than us (in particular omitting post-conditions), [Aucher and Schwarzentruber, 2012] shows that the model checking

problem in this case is PSPACE-Complete. In all likelihood this means we could do better than exponential blowup induced by tr, but we do not investigate this further here.

In certain cases it becomes relevant to our analysis whether the preconditions and postconditions of an event model are members of $\mathcal{L}_{\mathrm{EL}}(P)$ or $\mathcal{L}_{\mathrm{DEL}}(P)$. We therefore say that $\mathcal{E} = (E, \sim, pre, post)$ of $\mathcal{L}_{\mathrm{DEL}}(P)$ is *static* if for all $e \in E, p \in P$ we have that $pre(e) \in \mathcal{L}_{\mathrm{EL}}(P)$ and $post(e)(p) \in \mathcal{L}_{\mathrm{EL}}(P)$. We point out that in many cases of modelling epistemic planning (indeed every example we provide), static event models are ample enough to describe the actions available to an agent.

Proposition 3.16. *Given an epistemic model $\mathcal{M} = (W, \sim, V)$ on P and event model $\mathcal{E} = (E, \sim, pre, post)$. If $\mathcal{E}$ is static there exists an algorithm for computing $\mathcal{M} \otimes \mathcal{E}$ whose running time is polynomial in $|\mathcal{M}| \cdot |\mathcal{E}|$. Otherwise there exists an algorithm for computing $\mathcal{M} \otimes \mathcal{E}$ whose running time is polynomial in $|\mathcal{M}| \cdot 2^{|\mathcal{E}|}$.*

Proof. For each $(w, e) \in W \times E$ we need to determine $\mathcal{M}, w \models pre(e)$, and $\mathcal{M}, w \models post(e)(p)$ for all $p \in P$. Recalling that $|P| \leq |\mathcal{M}|$ this requires deciding no more than $|\mathcal{M}|^2 \cdot |E|$ model checking problems. If $\mathcal{E}$ is static it follows from Lemma 3.15 that each problem requires time polynomial in $|\mathcal{M}|$ and $|\mathcal{E}|$, as $|pre(e)| < |\mathcal{E}|$ and $|post(e)(p)| < |\mathcal{E}|$. Otherwise, we have by Lemma 3.15 that each problem requires time polynomial in $|\mathcal{M}|$ and $2^{|\mathcal{E}|}$. The other operations for computing $\mathcal{M} \otimes \mathcal{E}$ require at most time polynomial in $|W \times E|$ when using well-known data structures, hence the result follows. □

3.1.5 Types of Planning Problems

Picking up the discussion of nondeterminism and observability that we touched upon in Chapter 2, we now formalize what these notions mean to us here. These concepts directly influence the computational complexity of the solution existence problem, as is shown in the sequel.

Definition 3.17. Let an information cell $\mathcal{M}$ and an action $\mathcal{E}$ be given s.t. $\mathcal{E}$ is applicable in $\mathcal{M}$. We say that $\mathcal{E}$ is *non-branching* in $\mathcal{M}$, if the bisimulation contraction of $\mathcal{M} \otimes \mathcal{E}$ is a single information cell, and that $\mathcal{E}$ is *branching* otherwise. We say that $\mathcal{E}$ is *fully observable* in $\mathcal{M}$, if every information cell in the bisimulation contraction of $\mathcal{M} \otimes \mathcal{E}$ is a singleton, and that $\mathcal{E}$ is said to be *partially observable* otherwise.

It should be clear that this definition is in accord with our presentation so far, although for technical reasons we need to consider contraction-minimal mod-

els. Lifting this to the level of planning problems is intuitively straightforward, we simply require that every action of a planning problem satisfies the above definition in any information cell reachable from the initial state.

Definition 3.18. Let a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi)$ be given. Let $S_0 = \{\mathcal{M}_0\}$ and for $i > 0$ inductively define S_i as *the set of information cells reachable using i actions*, formally given by

$$\bigcup_{\mathcal{M} \in S_{i-1}} \{\mathcal{M}' \text{ is an info. cell of } \mathcal{M} \otimes \mathcal{E} \mid \mathcal{E} \in \mathsf{A} \text{ is applicable in } \mathcal{M}\}$$

We say that $\mathcal{P}$ is *non-branching* (*fully observable*) if for all $i \in \mathbb{N}$, each $\mathcal{E} \in \mathsf{A}$ is non-branching (fully observable) in every $\mathcal{M}$ of S_i. Otherwise $\mathcal{P}$ is *branching* (*partially observable*). The *reachable information cells of* $\mathcal{P}$ is $\cup_{i \in \mathbb{N}} S_i$.

Returning for a brief moment to automated planning, we have that *classical planning* studies fully observable and non-branching problems, and that *conformant planning* deals with partially observable and non-branching problems. *Contingency planning*, *conditional planning* and *nondeterministic planning* are frequently used in connection with planning problems that are branching, and to this is typically added the degree of observability assumed in the given environment. As such each of the four types of planning problems defined here is directly inspired by a counterpart in automated planning.

An important remark about Definition 3.18 is that it is given at the global level. By this we mean that it looks at the interaction between the initial state and any sequence of actions. It is also possible — and arguably more interesting — to find restrictions on the local level, where we for instance look at each action in the action library in isolation.[4] We now provide a few such local level restrictions, denoting by $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ some planning problem.

Proposition 3.19. *If $\mathcal{M}_0$ is a singleton and each $\mathcal{E} \in \mathsf{A}$ is a singleton, then $\mathcal{P}$ is non-branching and fully observable.*

Proof. By the product update operation we have that if $\mathcal{M}$ contains a single world and $\mathcal{E}$ a single event s.t. $\mathcal{E}$ is applicable in $\mathcal{M}$, then $\mathcal{M} \otimes \mathcal{E}$ is a singleton. With this it is straightforward to show that $\mathcal{P}$ satisfies both properties of Definition 3.18 by induction on the set of information cells reachable using i actions. □

Proposition 3.20. *If $\mathcal{M}_0$ is a singleton and for each $\mathcal{E} = (E, \sim, pre, post)$ in A we have that $\sim$ is the identity relation on E, then $\mathcal{P}$ is fully observable.*

[4] Much research into classical planning is concerned with exactly this. There the aim is to find relaxations of planning problems that makes them tractable, allowing this relaxed version to be used for heuristic guidance during e.g. a state-space search.

Proof. By the product update operation we have that if $\mathcal{M}$ is a singleton and the epistemic relation of $\mathcal{E}$ is the identity with $\mathcal{E}$ applicable in $\mathcal{M}$, then each information cell of $\mathcal{M} \otimes \mathcal{E}$ is a singleton. A proof by induction is readily seen using this fact. □

Proposition 3.21. *If for each $\mathcal{E} = (E, \sim, pre, post)$ in A we have that $\sim$ is the universal relation on E, then $\mathcal{P}$ is non-branching.*

Proof. For an information cell $\mathcal{M}$ and action $\mathcal{E}$ whose epistemic relation is universal and which is applicable in $\mathcal{M}$, we have that $\mathcal{M} \otimes \mathcal{E}$ is an information cell. A proof by induction is easily formulated using this fact, and recalling that S_i is a set of information cells ($\mathcal{M}_0$ is an information cell). □

In the general case we have that $\mathcal{P}$ is branching and partially observable.

3.1.6 Alternating Turing Machines and Complexity Classes

Alternating Turing machines (ATMs) is a model of computation generalizing both deterministic and nondeterministic Turing machines, and which are attributed to the authors of [Chandra et al., 1981]. Just like a regular Turing machine, an ATM is given some input and, according to a given set of rules, manipulates symbols on a tape that is divided into cells. We call these symbols letters to avoid confusion with propositional symbols. Here we introduce the formulation of ATMs found in [Sipser, 2006].

Definition 3.22. We define an *alternating Turing machine (ATM)* as a tuple $T = (Q, \Sigma, \delta, q_0, l)$, where

- Q is a finite set of internal states,
- Σ is a finite alphabet not containing the *blank symbol* $\sqcup$,
- $\delta : Q \times (\Sigma \cup \{\sqcup\}) \to 2^{(Q \times (\Sigma \cup \{\sqcup\}) \times \{L,R\})}$ is a transition function, and each element in the codomain is a *transition*,
- $q_0 \in Q$ is an initial state, and
- $l : Q \to \{\mathsf{acc}, \mathsf{rej}, \forall, \exists\}$ is a labelling function.

We say that $q \in Q$ is an (*accepting*/*rejecting*/*universal*/*existential*) state if $l(q) = (\mathsf{acc}/\mathsf{rej}/\forall/\exists)$. A *halting* state is a state that is either accepting or rejecting. If for all $q \in Q, s \in \Sigma$ we have that $|\delta(q, s)| \leq 1$ then T is a *deterministic Turing machine (DTM)*.

For a string $\sigma \in \Sigma^*$, we let $\sigma(i)$ denote the i'th letter of σ, and use the convention that $\sigma(1)$ is the first letter in the string. The *length* of σ's is given by the number of letters in σ and is denoted $|\sigma|$. The *empty string* ϵ has length 0.

Definition 3.23. Let an ATM $T = (Q, \Sigma, \delta, q_0, l)$ be given. We say that $c = \sigma q \sigma'$ is a *configuration* of T when $q \in Q$ and $\sigma, \sigma' \in \Sigma^*$. When T is in the configuration $c = \sigma q \sigma'$ it means that T is in state q. Further, $\sigma\sigma'$ is currently on the tape of T, with σ being the string left of the head, $\sigma'(1)$ being below the head, and the remaining symbols in σ' being to the right of the head (i.e. the substring $\sigma'(2)\dots\sigma'(|\sigma'|)$) possibly followed by blanks. If q is a (halting/existential/universal) state then c is a *(halting/existential/universal) configuration*. The *starting configuration* of T on input string $\sigma'' \in \Sigma^*$ is $\epsilon q_0 \sigma''$, meaning that $\sigma''(1)$ is below the head.

We assume the reader is familiar with how a nondeterministic Turing machine (NTM) computes, namely that it may proceed according to several possibilities, determined by its configuration and the transition function. Because an ATM computes in exactly the same manner we forego the formalities here. If T can go from configuration c to c' in one step, we say that c' is a *successor* configuration (of c), or in short that c *yields* c'. We also make some typical assumptions that does not incur a loss of generality: A halting configuration has no successor configurations, there *is* a left-hand end of the tape (it's extensible only to the right), and T never tries to move its head off the left-hand end of the tape. The following definition is adapted from [Rintanen, 2005] and is suited for the proofs we give below.

Definition 3.24. Given an ATM T and a configuration $c = \sigma q \sigma'$ of T. We define the following for $x, y \in \mathbb{N}$.

- When $l(q) = \mathsf{acc}$ we say that c is 0-accepting,
- When $l(q) = \forall$ we say that c is x-accepting if for *every* successor configuration c' there is a $y < x$ s.t. c' is y-accepting,
- When $l(q) = \exists$ we say that c is x-accepting if for *some* successor configuration c' there is a $y < x$ s.t. c' is y-accepting, and
- Otherwise c is rejecting.

When there exists an $x \in \mathbb{N}$ s.t. c is x-accepting, we say that c is an *accepting configuration* (of T). If the starting configuration of T on an input string σ'' is accepting, we say that T *accepts* σ''.

It may be rewarding to view the computation of an ATM as a tree. Here the root of the tree is the starting configuration, leaves correspond to halting configurations, internal nodes correspond to either universal configurations or existential configurations, and edges indicate possible transitions. This illustrates further that an ATM without universal states is simply an NTM, and that for DTMs this tree is a path corresponding to a sequence of configurations.

The set of input strings accepted by a Turing machine T is the *language recognized by* T. A decision problem can be seen as a boolean function mapping inputs (problem instances) to either YES or NO. Given a decision problem, the corresponding language is the set of inputs for which the answer is YES. A Turing machine recognizing such a language is said to solve the decision problem.

We will consider Turing machines whose computational resources are bounded, with the usual suspects being time and space. The time complexity of an ATM T is the maximum number of steps that T uses on any computation branch; i.e. from the root to any leaf. The space complexity of an ATM T is the maximum number of tape cells that T scans on any computation branch. We let $\mathrm{ATIME}(f(n))$ denote the set of decision problems solvable by an ATM, whose time complexity is $O(f(n))$ on any input of length n. Similarly, $\mathrm{ASPACE}(f(n))$ denotes the set of decision problems solvable by an ATM whose space complexity is $O(f(n))$ on any input of length n. In the case of DTM's (where there is a single computation branch), we use $\mathrm{DTIME}(f(n))$ and $\mathrm{DSPACE}(f(n))$ analogously. For $f(n) \geq \log n$ we have that $\mathrm{ASPACE}(f(n)) = \mathrm{DTIME}(2^{O(f(n))})$ [Sipser, 2006, Theorem 10.21], and with all this in place we can now give the complexity classes we're concerned with here.

$$
\begin{aligned}
\mathrm{PSPACE} &= \cup_{0\leq k}\mathrm{DSPACE}(n^k) \\
\mathrm{EXP} &= \cup_{0\leq k}\mathrm{DTIME}(2^{n^k}) = \cup_{0\leq k}\mathrm{ASPACE}(n^k) = \mathrm{APSPACE} \\
\mathrm{EXPSPACE} &= \cup_{0\leq k}\mathrm{DSPACE}(2^{n^k}) \\
\mathrm{2EXP} &= \cup_{0\leq k}\mathrm{DTIME}(2^{2^{n^k}}) = \cup_{0\leq k}\mathrm{ASPACE}(2^{n^k}) = \mathrm{AEXPSPACE}
\end{aligned}
$$

Consider now C denoting one of the complexity classes above, $\mathcal{L}$ a decision problem in C and T a TM solving $\mathcal{L}$. In Section 3.4 we show how to construct a planning problem $\mathcal{P}$ whose size is polynomial in the size of T and an input string σ, s.t. T accepts σ iff there exists a solution to $\mathcal{P}$. In other words, to establish C-Hardness we directly simulate any Turing machine (decision procedure) that solves problems in C.

3.2 Suboptimality of StrongPlan($\mathcal{P}$)

Using the results of Chapter 2, we can easily give an effective (i.e. decidable) procedure for answering the solution existence problem for $\mathcal{P}$. Simply apply StrongPlan($\mathcal{P}$) and answer NO if the algorithm returns fail, and otherwise answer YES. By Theorem 2.26 it follows that the solution existence problem is decidable. However, this approach falls short for determining the computational complexity of the solution existence problem; it is suboptimal.

Given a planning problem $\mathcal{P}$ on P, we can take on a general perspective and consider StrongPlan($\mathcal{P}$) as a tree search algorithm. This means the number of nodes expanded is $O(b^d)$ where b is the branching factor and d the maximum depth of the tree.

As far as $\mathcal{B}_2$-saturated planning trees go, we have that an or-node has at most one child per action in the action library. Further, the number of children of an and-node is at most the number of connected components in an action (i.e. the number of equivalence classes under the epistemic relation of the event model). Therefore the branching factor b is the greater number of either of these two quantities. Turning to the depth of a planning tree, we have that on any path from the root to a leaf, at most two or-nodes are totally bisimilar due to to blocking condition $\mathcal{B}_2$.[5] Additionally, there is one more or-node than there are and-nodes on such a path due to the tree expansion rule. Consequently the upper bound on d is one, plus twice the number of non-totally bisimilar information cells on P.

These observations together with Lemma 3.12 means we have $O(b^{(2^{2^n+1})})$ as an upper bound on the worst case running time of StrongPlan($\mathcal{P}$) for $|P| = n$. When $\mathcal{P}$ is fully observable (every reachable information cell is a singleton) the upper bound is instead $O(b^{2^n+1})$ due to Lemma 3.10.

We now give a concrete example showing that planning trees can grow inconveniently large, in spite of the restrictions imposed by blocking condition $\mathcal{B}_2$. We will say that an epistemic model *satisfies exactly* p when p holds in every world of an epistemic model, and every symbol in $P \setminus \{p\}$ does not hold.

Definition 3.25. Given some $k \in \mathbb{N}$ let $P = \{p_1, \ldots, p_k, p_{k+1}, p_{k+2}\}$ be a set of $k+2$ propositional symbols. We define the planning problem $\mathcal{P}^k = (\mathcal{M}_0, \mathsf{A}, p_{k+2})$ on P, where

- $\mathcal{M}_0$ is a singleton epistemic model satisfying exactly p_1, and

[5] Formally, it is the respective labels of nodes that denote bisimilar epistemic models, but we will allow ourselves this abuse for readability.

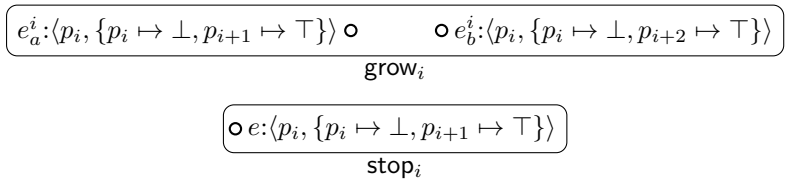


Figure 3.1: Event models used in $\mathcal{P}^k$.

- $\mathsf{A} = \{\mathsf{grow}_1, \ldots, \mathsf{grow}_k, \mathsf{stop}_{k+1}\}$ is the action library where each event model is as indicated in Figure 3.1 (for $k = 0$ we have $\mathsf{A} = \{\mathsf{stop}_1\}$).

Using Lemma 3.20 we see that $\mathcal{P}^k$ is a branching and fully observable planning problem.

Example 3.26. In Figure 3.2 we've illustrated a planning tree T for $\mathcal{P}^4$ that has been partially expanded as per the tree expansion rule. In the illustration doubly drawn nodes are solved, OR-nodes are represented by a circle and AND-nodes by a gray square. For each OR-node n in T its label $\mathcal{M}(n)$ is a singleton epistemic model which satisfies exactly one of the propositional symbol in $\{p_1, \ldots, p_6\}$. For visual simplicity we indicate only these symbols in our illustrations, and further we leave it to the reader to deduce the epistemic models labelling AND-nodes by inspecting their children.

We see that AND-nodes expanded using grow_i consists of two information cells, while AND-nodes expanded using stop_5 consists of one information cell. In this illustration there are 4 nodes in T which may be chosen for expansion (each leaf not satisfying p_6), as there are no applicable actions in the other OR-nodes. No node is prevented from expansion due to $\mathcal{B}_2$.

To show the general behaviour of StrongPlan($\mathcal{P}^k$) we need to determine exactly when the planning tree is $\mathcal{B}_2$-saturated. This is so when no OR-node can be chosen for expansion, making the blocking condition $\mathcal{B}_2$ superfluous for this particular planning problem. Intuitively, this is the case because nodes not satisfying the goal formula can only be expanded by a single action, and nodes that *do* satisfy the goal formula cannot be expanded at all. Moreover, no path from the root to a leaf contains two OR-nodes satisfying the same symbols, hence they cannot be bisimilar. For brevity we have therefore put proof of the following result in the appendix.

Lemma 3.27. *$\mathcal{B}_2$ does not prevent the tree expansion rule from being applied to any* OR*-node n of a planning tree T for $\mathcal{P}^k$.*

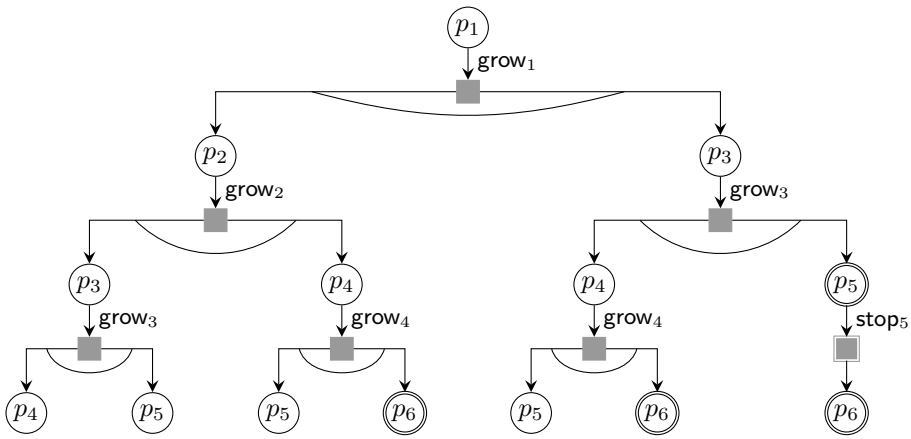


Figure 3.2: Partially expanded planning tree for $\mathcal{P}^4$

Proof. See Lemma A.5 □

What Lemma 3.27 tells us is that regardless of the non-deterministic choice made in the tree expansion rule, all $\mathcal{B}_2$-saturated planning trees for $\mathcal{P}^k$ are isomorphic, because such trees will have had every possible node expanded. Therefore StrongPlan($\mathcal{P}^k$) always runs until there are no more nodes left to expand, that is, until every leaf satisfies p_{k+2}. The running time of StrongPlan($\mathcal{P}^k$) is therefore proportional to the number of nodes in a fully expanded tree, and we now show this to grow exponentially in k.

From Example 3.26 and the (somewhat) suggestive naming, the very astute reader may by now have recognized that any $\mathcal{B}_2$-saturated planning tree for $\mathcal{P}^k$ is reminiscent of a *Fibonacci tree*. We therefore recall some facts about the famed Fibonacci numbers. We let F_j denote the j'th Fibonacci number, and recall the two identities $F_j = F_{j-1} + F_{j-2}$ and $\sum_{j=1}^{l} F_j = F_{l+2} - 1$. Moreover, a closed-form expression for F_j is $\left[\frac{\varphi^j}{\sqrt{5}}\right]$ where $[\cdot]$ is the *nearest integer function* and $\varphi > 1.61$ is the golden ratio.

Proposition 3.28. *Let $k \in \mathbb{N}$ and consider any $\mathcal{B}_2$-saturated planning tree T for $\mathcal{P}^k = (\mathcal{M}_0, \mathsf{A}, p_{k+2})$. The number of nodes in T is exponential in k.*

Proof. By Lemma 3.27 we have that T is $\mathcal{B}_2$-saturated exactly when every node has been expanded, so we will first proceed to count the number of OR-nodes in a fully expanded planning tree T. We'll refer to an OR-node n in T as a p_i-node when it is the case that $\mathcal{M}(n)$ satisfies exactly p_i, and further let $|p_i|$ denote the number of p_i-nodes in T.

Since no action adds a p_1-node, we have immediately from the initial state that $|p_1| = 1$. We now settle the number of OR-nodes resulting from applying the tree expansion rule in some p_i node. When stop_{k+1} is applicable the result is a p_{k+2}-node, and furthermore for $1 \leq i \leq k$ we have that when grow_i is applicable, the result is a p_{i+1}-node and a p_{i+2}-node. To determine $|p_2|$, we have in the case of $k = 0$ that $|p_2| = 1$ due to stop_1. Otherwise when $k > 0$, then we have from grow_1 that $|p_2| = |p_1| + |p_0|$ which equals 1 as there are no p_0-nodes. Summing up we have that $|p_1| = 1$ and $|p_2| = 1$ for any k.

For $3 \leq j \leq k+2$ we have that one p_j-node is added for each p_{j-1}-node and one for each p_{j-2}-node; with the case of $j = k + 2$ combining the single node from grow_k and the single node from stop_{k+1}. Put on a symbolic form this means $|p_j| = |p_{j-1}| + |p_{j-2}|$. Observe now that $|p_1| = F_1$, $|p_2| = F_2$, and since $|p_j|$ is identical to the Fibonnaci recurrence we arrive at $|p_j| = |p_{j-1}| + |p_{j-2}| = F_{j-1} + F_{j-2} = F_j$.

From Lemma A.1 we have that every OR-node in T satisfies exactly one propositional symbol. Therefore the total number of OR-nodes in T is precisely given by $|p_1| + \cdots + |p_{k+2}| = F_1 + \cdots + F_{k+2} = F_{k+4} - 1 = \left[\frac{\varphi^{k+4}}{\sqrt{5}}\right] - 1$. Since T is a planning tree there is at least one more OR-node than there are AND-nodes, hence the number of AND-nodes in T is at most $F_{k+4} - 2$, thus we have

$$\left[\frac{\varphi^{k+4}}{\sqrt{5}}\right] - 1 \leq |T| \leq 2\left[\frac{\varphi^{k+4}}{\sqrt{5}}\right] - 3 \qquad \text{as } \varphi > 1 \Rightarrow$$
$$|T| = \Theta(c^k) \qquad \text{for some } c > 1$$

, thereby showing that the number of nodes in T is exponential in k. □

Corollary 3.29. StrongPlan *is an exponential time algorithm for the solution existence problem for* $\mathcal{P}^k$.

In an alternative formulation of $\mathcal{P}^k$ we can consider $\{p_1, \ldots, p_{k+2}\}$ as encoding an integer, and construct actions that capture integer arithmetic — we use such actions in Section 3.4.3. A $\mathcal{B}_2$-saturated planning tree for this alternative version contains $F_{2^{k+2}+2}$ OR-nodes, and so the running time of StrongPlan is in this case doubly exponential in k, coinciding roughly with the upper bound for fully observable planning problems that we presented at the start of this section. As shown later in Theorem 3.53, this implies that StrongPlan is asymptotically suboptimal for deciding whether a solution exists.

3.3 Optimal Decision Procedures

In this section we introduce asymptotically optimal decision procedures for answering the solution existence problem, with optimality being shown by the results of Section 3.4. We put the tools of Section 3.1 to use here, both for showing correctness of this procedure and for bounding the resources used. The upper bounds differ for each of the four types of planning problems mentioned in Section 3.1.5. For planning problems that are branching we replace planning trees with planning graphs, so that our algorithm can avoid expanding redundant nodes. Non-branching problems are simpler because we only need to determine whether a *sequential* plan exists, and so we can use a strategy that requires fewer space resources in this case.

3.3.1 Replacing Planning Trees

As illustrated in Section 3.2 we risk adding an exponential number of bisimilar nodes to the planning tree when using this as our underlying data structure. We can eliminate this behaviour from our decision procedure by identifying nodes that, loosely stated, contain the same information. Foregoing the extraction of plans from planning trees, the nodes we seek to identify are those that "behave the same" when considering in tandem the tree expansion rule (Definition 2.18) and the notion of solved nodes (Definition 2.21). This means that for two nodes n, n' to be identified we must have that $\mathcal{M}(n)$ and $\mathcal{M}(n')$ are modally equivalent (Definition 3.3) and further, if we take any event model $\mathcal{E}$, the nodes added by the tree expansion rule using n and $\mathcal{E}$ are modally equivalent to the nodes added using n' and $\mathcal{E}$. To incorporate this fact into StrongPlan one might attempt to augment $\mathcal{B}_2$ further and keep employing planning trees. However, a more rewarding approach is to do away with trees as whole, and instead turn to a more suited form of graph.

Definition 3.30. A *planning graph* $G = (N, M, E, n_0)$ is a bipartite, rooted, directed and labelled graph. N and M are disjoint sets of nodes, where N denotes a set of OR-nodes and M a set of AND-nodes. The OR-node $n_0 \in N$ is the *root*, also denoted $root(G)$. Each node v of G is labelled by an epistemic model $\mathcal{M}(v)$. Each edge $(n, m) \in E$ with $n \in N$ and $m \in M$ is labelled by an event model $\mathcal{E}(n, m)$. We require that there is a path from n_0 to every other node in G (implying that G is weakly connected).

Given a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$, a planning graph $G = (N, M, E, n_0)$ for $\mathcal{P}$ is one in which $\mathcal{M}(n_0) = \mathcal{M}_0$ and where every edge (n, m) from an OR-node to an AND-node has $\mathcal{E}(n, m) \in \mathsf{A}$. The *initial planning graph* for $\mathcal{P}$ is

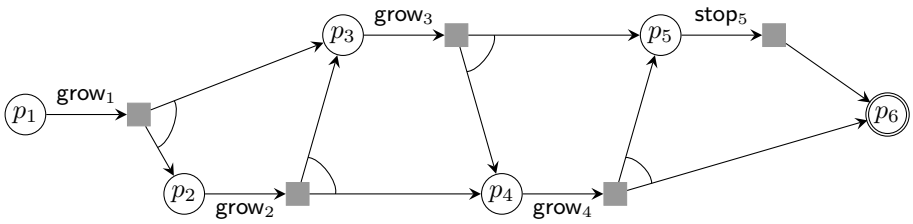


Figure 3.3: Saturated planning graph for $\mathcal{P}^4$

$G_0 = (\{n_0\}, \emptyset, \emptyset, n_0)$ where $\mathcal{M}(n_0) = \mathcal{M}_0$. For $n \in N$ and $\mathcal{E} \in \mathsf{A}$, when $\mathcal{E}$ is applicable in $\mathcal{M}(n)$ and no edge leaving n is labelled by $\mathcal{E}$, we say that n is *valid (for expansion)* with $\mathcal{E}$.

Definition 3.31. Let a planning graph $G = (N, M, E, n_0)$ for $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ be given. For $n \in N$ and $\mathcal{E} \in \mathsf{A}$ s.t. n is valid for expansion with $\mathcal{E}$, the *graph expansion rule* extends G by:

1. Adding a new node AND-node m to M with $\mathcal{M}(m) = \mathcal{M}(n) \otimes \mathcal{E}$, and adding the edge (n, m) with $\mathcal{E}(n, m) = \mathcal{E}$.
2. For each information cell $\mathcal{M}'$ in $\mathcal{M}(m)$:
   (a) If there is an OR-node $n' \in N$ s.t. $\mathcal{M}(n') \leftrightarrows \mathcal{M}'$, then add the edge (m, n'),
   (b) Otherwise add a new OR-node n' to N with $\mathcal{M}(n') = \mathcal{M}'$ and add the edge (m, n').

When G has been expanded by the above rule and no choice of n and $\mathcal{E}$ is valid we say that G is *saturated*.

Note that a saturated planning graph is finite, because there are finitely many non-totally bisimilar information cells and finitely many event models in the action library. Both Definition 3.30 and 3.31 are closely connected to their counterparts in Section 2.4, and as such planning graphs are a natural development of planning trees. A key difference is that no two OR-nodes in a planning graph are bisimilar as a consequence of (2.a) in the graph expansion rule. This was handled differently using blocking conditions in the previous. Less relevant is that we, for simplicity, define a planning graph as saturated once every reachable information cell of the planning problem has been expanded.

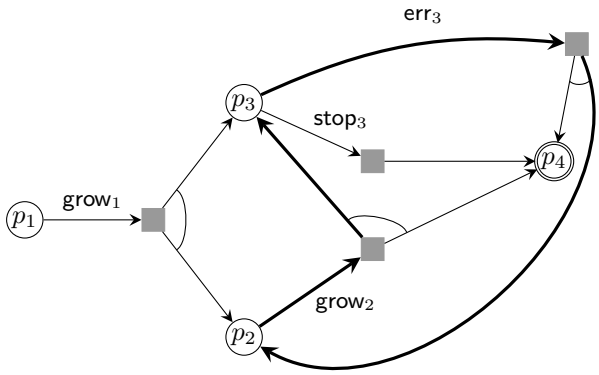


Figure 3.4: Saturated planning graph for $\mathcal{P}^2_{\text{ext}}$, cycle indicated with thick arrows.

Example 3.32. We've illustrated a saturated planning graph for $\mathcal{P}^4$ in Figure 3.3. As the graph expansion rule is applied, we check in (2.a) for each information cell resulting from the product update, whether the planning graph already contains an OR-node whose label is bisimilar to the information cell. The effect is that we only add 6 OR-nodes to the planning graph, which is in stark contrast to the $F_8 - 1$ OR-nodes we would add using planning trees and the tree expansion rule. More generally, for $\mathcal{P}^k$ we have that the size of the saturated planning graph is polynomial k. Observe that the graph is acyclic and each OR-node has either out-degree 1 or satisfies p_6. Therefore if we consider the direct successors (in a graph) as the counterpart to the children (in a tree), we can intuitively see how a plan could be extracted in a manner similar to Definition 2.22.

Example 3.33. Consider the planning problem $\mathcal{P}^2_{\text{ext}}$, which modifies $\mathcal{P}^k$ by adding an action err_3. err_3 is applicable in information cells satisfying p_3, and results in an epistemic model containing two information cells; one satisfying exactly p_2 and the other satisfying exactly p_4. The saturated planning graph for this problem is illustrated in Figure 3.4 (it should be clear how err_{k+1} for $\mathcal{P}^k_{\text{ext}}$ would be similarly defined). The addition of err_3 leads to a cycle (indicated with thick arrows) in the saturated planning graph, so we might cycle between information cells satisfying exactly p_2 and p_3 if we execute grow_2 in the former and err_3 in the latter. At the same time, any solution to $\mathcal{P}^2$ is also a solution to $\mathcal{P}^2_{\text{ext}}$ as adding actions to the action library cannot make a problem unsolvable. As the p_3-node now has an outdegree of 2 it is however less obvious, in contrast to Example 3.32, on which grounds we should preclude err_3 from being part of a solution.

To handle the general case, we must be able to determine whether a planning graph admits a solution when it is cyclic and the outdegree of OR-nodes is greater than 1. To this end we will look for an acyclic subgraph satisfying certain criteria that guarantees the existence of a strong solution.

Definition 3.34. Let a saturated planning graph $G = (N, M, E, n_0)$ for $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ be given. A *solved subgraph of* G *for* $n'_0 \in N$ is a subgraph $G' = (N', M', E', n'_0)$ of G satisfying the following criteria.

$\mathcal{C}_1$ n'_0 is the single source node (indegree is 0), and every node in G' is reachable from n'_0,

$\mathcal{C}_2$ G' is acyclic,

$\mathcal{C}_3$ For each $n' \in N'$: the outdegree of n' is 0 if $\mathcal{M}(n') \models \phi_g$ (a goal node is a sink), and otherwise the outdegree of n' is 1 and this outgoing edge has the same label as in G,

$\mathcal{C}_4$ For each $m' \in M'$: $(m', n') \in E'$ iff $(m', n') \in E$ (outgoing edges of AND-nodes are exactly as in G).

We say that G is *solved* if there exists a solved subgraph of G for n_0.

Observe that n'_0 can be any OR-node of the saturated planning graph. We also require the following intuitive definition used in the results shown in Appendix 3.3.

Definition 3.35. Given a solved subgraph $G' = (N', M', E', n'_0)$, we define for any node v' of G' its *height* $h(v')$ recursively by:

- If $v' \in N'$ and $\mathcal{M}(v') \models \phi_g$ then $h(v') = 0$,
- If $v' \in N'$ and $\mathcal{M}(v') \not\models \phi_g$, then $h(v') = h(m') + 1$ where m' is the single the direct successor of v', and
- If $v' \in M'$, then $h(v') = \max(h(n'_1), \ldots, h(n'_k)) + 1$ where $n'_1, \ldots, n'_k$ are the direct successors of v'.

From $\mathcal{C}_2$ we have that a solved subgraph G' is an acyclic subgraph G and so G' contains no paths of infinite length. $\mathcal{C}_3$ means we assign to each reachable information cell (not satisfying the goal) a single action, whose expansion is also contained in G' by virtue of $\mathcal{C}_4$, hence G' faithfully represents the graph expansion rules applied when saturating G. Taking into account $\mathcal{C}_1$, G' fully

describes the actions an agent must take from the initial state in order to reach a sink, regardless of what contingency arise during execution. In this sense a solved subgraph contains all the information necessary in order to synthesise a plan, and as such plays the same role as a planning tree whose root is solved.

3.3.2 Working with Planning Graphs

In showing correctness of the decision procedure we develop in Section 3.3.3, we first show that notion of a solved planning tree coincides with that of a solved subgraph.

Lemma 3.36. *Let a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ be given. There exists a $\mathcal{B}_2$-saturated planning tree for $\mathcal{P}$ whose root is solved iff the saturated planning graph G for $\mathcal{P}$ is solved.*

Proof. Assume that we're given a $\mathcal{B}_2$-saturated planning tree T for $\mathcal{P}$ such that $root(T)$ is solved. We must show that the saturated planning graph for $\mathcal{P}$ is solved. We outline how this is shown, with a full proof of this direction being available in the proof of Lemma A.6 in Appendix A.2. The proof proceeds by constructing from T a graph G' which constitutes a solved subgraph of G *up to bisimilarity*. The first step is pruning T so that it contains only solved nodes, which implies that OR-nodes in T have outdegree 0 if they satisfy ϕ_g and otherwise they have outdegree 1. Starting from the leaves of T, we iteratively move up the planning tree, extending G' with the OR-nodes and AND-nodes encountered with the proviso that no two bisimilar OR-nodes are added to G'. In this way, the children of any node we process in T are guaranteed to (at least) have a bisimilar partner in G', allowing us construct G' so that the edges in T are matched up to bisimilarity.

With a few modifications, we can map G' to a subgraph of G which satisfies $\mathcal{C}_1$ and $\mathcal{C}_2$. That this mapping satisfies $\mathcal{C}_3$ follows from the initial pruning of T. Showing this mapping satisfies $\mathcal{C}_4$ is done on the basis that both the tree expansion rule and graph expansion rule use the product update operation. This implies the existence of a solved subgraph of G for $\mathcal{P}$ thereby concluding this direction.

We only sketch the other direction, whose proof is conceptually more simple. It is essentially the inverse of the procedure we gave above, that is, we take a solved subgraph of G, and from this construct a planning tree (up to bisimilarity) whose root is solved. Assuming that we're given a solved subgraph G' of G, we manipulate it by repeatedly splitting an OR-node n whose indegree is greater than 1, until no such n exists. When splitting a node, we add a new node n'

whose label is $\mathcal{M}(n)$, and one incoming edge of n is modified so that it instead points to n'. Below n' we add a copy of the acyclic subgraph that has n as its root. As G' initially satisfies $\mathcal{C}_2$ we have that the split operation preserves acyclicity. Furthermore, every sink of G' remains a sink and the source is never split as its indegree is 0. Therefore once every OR-node in G' has indegree at most 1, it constitutes a planning tree (up to bisimilarity) whose root is the root of G', and whose leaves are the sinks in G' meaning that each leaf satisfies ϕ_g. Again, the correspondence between the graph expansion rule and the tree expansion rule means this procedure in fact gives a $\mathcal{B}_2$-saturated planning tree for $\mathcal{P}$. □

Having established the result above, we are still in need of a procedure for determining whether a saturated planning graph is solved. To this end, we adapt the algorithm StrongPlan found in [Cimatti et al., 2003] to our setting. While StrongPlan assumes a fully observable domain, the extension to partial observability is straightforward. Roughly speaking, StrongPlan starts from an empty plan, and in each iteration of its main loop extends this plan by assigning actions to states. There are two requirements for a state to be assigned an action. First, a state is at most assigned one action (ensured by PruneStates). Second, the action must guarantee that its outcomes are either goal states or states for which an action has already been assigned (achieved using StrongPreImage). Keeping in mind these two requirements, we now introduce the procedure SolvedSubgraph.

Definition 3.37. Let a saturated planning graph $G = (N, M, E, n_0)$ be given. For any $N' \subseteq N$, we define the predicate $Solved(m, G, N') \Leftrightarrow (\forall (m, n) \in E : n \in N')$.

Informally we have that $Solved(m, G, N')$ holds if an action that produces m is guaranteed to lead to one of the information cells in N'. As such, *Solved* serves to determine whether the second requirement mentioned above holds in our setting, and is directly put to work in line 6 of SolvedSubgraph. The first requirement is achieved in line 4 where we ignore any OR-node already contained in G'. The input of SolvedSubgraph is a saturated planning graph and a goal formula.

SolvedSubgraph$(G = (N, M, E, n_0), \phi_g)$
1 $N' = \{n \in N \mid \mathcal{M}(n) \models \phi_g\}$, $M' = \emptyset$, $E' = \emptyset$
2 $G' = (N', M', E')$
3 **while** G' is modified by the following
4 $N_{\text{us}} = N \setminus N'$ ▷ Unsolved OR-nodes
5 **for** each $n \in N_{\text{us}}$
6 **if** n has a direct successor m in G s.t. $\mathit{Solved}(m, G, N')$ **then**
7 Add n, m, (n, m) and $(m, n_1), \ldots, (m, n_k)$ to G'
where and $n_1, \ldots, n_k$ are the direct successors of m in G.
▷ N', M', E' of G' are modified
8 **return** G'

The following result is oblivious to the actual cost of the operations in SolvedSubgraph, meaning that we simply count the number of high-level steps described. We rectify this shortcoming later when it becomes prudent to our analysis.

Lemma 3.38. *The number of steps used in* SolvedSubgraph *is polynomial in* $|N|$.

Proof. We count the number of iterations in the nested loop construction. For each iteration starting at line 3, we have that the loop ends unless G' is modified in line 7. Each iteration modifying G' adds at least one node in N to N'. Therefore the number of iterations modifying N' is at most the number of nodes in N. Furthermore the loop in line 5 requires at most N iterations, and so line 7 is reached at most $|N|^2$ times. □

We will use G_i to denote the graph produced after i iterations of the loop in line 3 of SolvedSubgraph. Observe that as SolvedSubgraph never removes nodes, we have that G_i is a subgraph of G_{i+1}. The following result essentially states that each OR-node n^* we add to G' in line 7 can be composed with subcomponents of G' to form a solved subgraph of G for n^*.

Lemma 3.39. *Given a saturated planning graph* $G = (N, M, E, n_0)$ *for* $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$*, and let* $G_i = (N_i, M_i, E_i)$*. Further, let* $n^* \in (N \setminus N_i)$ *s.t.* n^* *has a direct successor* m^* *in* G *with* $\mathit{Solved}(m^*, G, N_i)$*, and where* $n_1, \ldots, n_y \in N_i$ *denotes the direct successors of* m^* *(see also Figure 3.5). If for each* x *s.t.* $1 \leq x \leq y$ *there exists a solved subgraph of* G_i *for* n_x*, then there exists a solved subgraph of* G *for* n^*.

Proof. We form the graph G^* that is indicated in Figure 3.5. We then show that this graph satisfies the four criteria for being a solved subgraph of G for n^*. A full proof is given when showing Lemma A.8 in Appendix A.2.

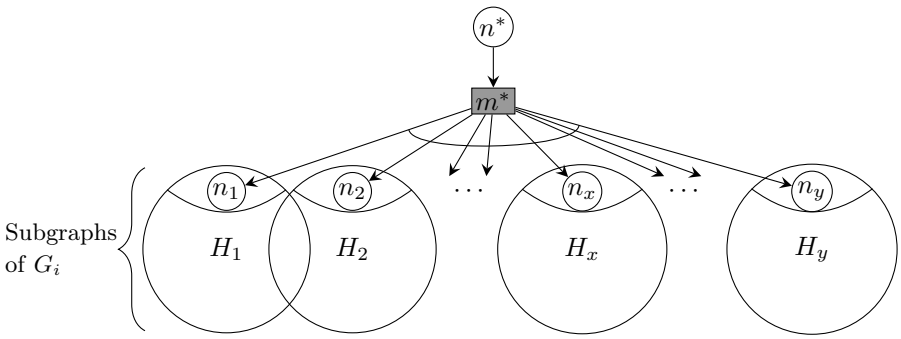


Figure 3.5: Illustration of how G^* as used in Lemma 3.39 is formed. Observe that the node set of any of the composed graphs (except for their roots) may overlap — not just H_1 and H_2.

□

We now show that any OR-node added to G' by SOLVEDSUBGRAPH is solved, which is the last result necessary in order to show correctness of the ensuing decision procedure.

Lemma 3.40. *Let a saturated planning graph $G = (N, M, E, n_0)$ for $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ be given, and let $G' = \textsc{SolvedSubgraph}(G, \phi_g)$. For each $n \in N$, there exists a solved subgraph G_s of G for n iff n belongs to G'.*

Proof. We show first that if there exists a solved subgraph G_s of G for n, then n belongs to G'. Let $h(n) = 2j$ for $j \geq 0$ denote the height of n in G_s according to Definition 3.35, and assume towards a contradiction that n does not belong to G'. We proceed by induction on j. For the base case $j = 0$ and so $h(n) = 0$, meaning that $\mathcal{M}(n) \models \phi_g$. In line 2 of SOLVEDSUBGRAPH each node satisfying ϕ_g is added to G', and as nodes are never removed this contradicts that n does not belong to G'.

For the induction step assume that $h(n) = 2(j+1) = 2j+2$. From $\mathcal{C}_1$ we have that there is a path to every node in G_s from n, and so it must be that every other node in G_s has height $\leq 2j+1$. By $\mathcal{C}_3$ we have that n has a single direct successor m, and by $\mathcal{C}_4$ that m has $n_1, \ldots, n_y$ direct successors, each of which therefore has height at most $2j$. Therefore we can apply the induction hypothesis to conclude that each $n_1, \ldots, n_y$ belong G'. By Lemma 3.38 we have that SOLVEDSUBGRAPH terminates, this means there is some $i \in \mathbb{N}$ s.t. each $n_1, \ldots, n_y$ belong to G_i (the graph produced after i iterations of the loop in line 3). As we assume n does not belong to G' it cannot belong to G_i either, hence

$n \in (N \setminus N_i)$. But we now have $n \in N_{\text{us}}$ and that n and its direct successor m satisfy the conditions condition in line 6 of SolvedSubgraph, and so it follows that n belongs to G_{i+1}, thus contradicting that n does not belong to G'.

Turning to the converse, we now show that if n belongs to G', then there exists a solved subgraph G_s of G for n. We proceed by induction on the number i of iterations by the loop in line 3. For the base case assume that 0 iterations have been made, which implies that $n \in \{n' \in N \mid \mathcal{M}(n') \models \phi_g\}$. Therefore $\mathcal{M}(n) \models \phi_g$, and so $(\{n\}, \emptyset, \emptyset, n)$ is a solved subgraph of G for n. For the induction step assume that $i+1$ iterations has been made. If n was added in iteration $\leq i$, then the induction hypothesis immediately gives that there exists a solved subgraph for n. We can therefore assume that n was added in iteration $i+1$, meaning that $n \in (N \setminus N_i)$. From line 6 it follows that n has a direct successor m in G with $Solved(m^*, G, N_i)$, and where $n_1, \ldots, n_y \in N_i$ are the direct successors of m^*. For $1 \leq x \leq y$ we have that n_x was added in an iteration $\leq i$, and so by the induction hypothesis there exists a solved subgraph H_x of G_i for n_x. This means we can apply Lemma 3.39 to conclude that there exists a solved subgraph of G for n, thus concluding the induction step. □

3.3.3 Deciding Whether a Solution Exists

We present an optimal decision procedure for the solution existence problem, and show that this procedure is both sound and complete.

SolutionExists($\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$)

1 Construct the initial planning graph $G = (\{n_0\}, \emptyset, \emptyset, n_0)$ for $\mathcal{P}$ with $\mathcal{M}(n_0) = \mathcal{M}_0$.
2 Repeatedly apply the graph expansion rule until G is saturated, with the addition that labels of OR-nodes are bisimulation contracted.
3 Answer YES if n_0 belongs to SolvedSubgraph(G, ϕ_g), otherwise answer NO.

In line 2 of SolutionExists we make sure that the labels of OR-nodes in the planning graph represent contraction-minimal information cells. Doing this incurs little overhead when constructing the saturated planning graph, and ensures that the epistemic models are of reasonable size.

Theorem 3.41. SolutionExists *is a sound and complete decision procedure for the solution existence problem.*

Proof. Soundness means that if SolutionExists($\mathcal{P}$) answers YES, then there exists a solution to $\mathcal{P}$. When n_0 belongs to SolvedSubgraph(G, ϕ_g), then it

follows from Lemma 3.40 that G is solved. By Lemma 3.36 we therefore have that there exists a planning tree for $\mathcal{P}$ whose root is solved. Applying Theorem 2.24 it follows that there exists a solution to $\mathcal{P}$ as required. To show completeness, assume that there exists a solution to $\mathcal{P}$. By Theorem 2.25 this implies the existence of $\mathcal{B}_2$-saturated planning tree for $\mathcal{P}$ whose root is solved. By Lemma 3.36 we therefore have that G is solved, and applying Lemma 3.40 it follows that n_0 belongs to SolvedSubgraph(G, ϕ_g). Therefore SolutionExists answers YES as required. □

We now make our analysis of the running time of SolutionExists more concrete, namely as a function of the size of the input planning problem. For a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ its size is given in the obvious way, viz. $|\mathcal{P}| = |\mathcal{M}_0| + \sum_{\mathcal{E} \in \mathsf{A}} |\mathcal{E}| + |\phi_g|$ (see Section 3.1.3 for more on the size of inputs).

Theorem 3.42. *The solution existence problem is in* 2EXP.

Proof. Given any planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ on P we show that the running time of SolutionExists$(\mathcal{P})$ is $O(2^{2^{ik}})$ where $i = |\mathcal{P}|$ and k is some constant. From Lemma 3.11 the number of worlds in any contraction-minimal information cell on P is at most $2^{|P|}$. As $|\mathcal{M}_0| + |P| \leq i$ this means any information cell of an OR-node in the saturated planning graph has size at most $O(2^i)$.

We now consider the operations of the algorithm whose running time depends on the size of the input, namely graph expansion and bisimulation contraction. When applying the graph expansion rule we need to determine applicability, a model checking task, and compute the product update. Important here is that the size of any precondition and any event model is at most i. Using Proposition 3.15 and Proposition 3.16 and the fact that any information cell has size at most $O(2^i)$, this means we can compute the application of any graph expansion rule using time polynomial in 2^i. For bisimulation contraction, we have by Lemma 3.9 a simple characterization of bisimilar worlds in an information cell. Therefore, a contraction-minimal model can be obtained by removing worlds of an information cell until no two have the same label. From $|\mathcal{M}_0| \leq i$ and the fact that OR-nodes are bisimulation contracted immediately, computing the bisimulation contraction of any information cell can be done in time polynomial in 2^i.

By definition of the graph expansion rule, no two OR-nodes in the planning graph are bisimilar and furthermore each OR-node is expanded at most once for each action in the action library. With the number of non-bisimilar information cells on P being at most 2^{2^i} by Lemma 3.12, it follows that the number of OR-nodes in the saturated planning graph is at most 2^{2^i}. Moreover, since $|\mathsf{A}| \leq i$ and

each expansion requires time polynomial in 2^i, this means there is a constant k_1 such that $O(2^{(2^i)^{k_1}})$ is an upper bound on the time used for constructing the saturated planning graph in line 2 of SolutionExists.

The initial computation of nodes satisfying ϕ_g (line 1 of SolvedSubgraph) requires solving at most 2^{2^i} model checking tasks. The input of each task is a contraction-minimal information cell of size at most $O(2^i)$ and a formula of size at most i. The remaining steps of SolvedSubgraph can be computed in time polynomial in the size of the saturated planning graph by Lemma 3.38, and so there are at most 2^{2^i} such steps. From these facts and Proposition 3.15 it follows that there is exists a constant k_2 such that $O(2^{(2^i)^{k_2}})$ is an upper bound on the time required for computing line 3 of SolutionExists.

Consequently we have that the running time of SolutionExists($\mathcal{P}$) is

$$O(2^{(2^i)^{k_1}} + 2^{(2^i)^{k_2}})$$

. Letting $k = max(k_1, k_2)$ this reduces to $O(2^{2^{i^k}})$, and by Theorem 3.41 we therefore have that the solution existence problem is in 2EXP. □

It is worth pointing out that it is the number of propositions in P which is the most significant factor in the asymptotic analysis of SolutionExists. Moreover, this analysis also reveals that it is the number of non-totally bisimilar information cells that prevents a better upper bound on the running time of SolutionExists. In fully observable planning problems we have that any reachable information cell is a singleton. This means that the number of non-totally bisimilar information cells for such planning problems is at most 2^i by Lemma 3.10. From these facts we can readily establish the following result.

Theorem 3.43. *The solution existence problem for fully observable planning problems is in* EXP.

[Littman, 1997] shows that the solution existence problem for fully observable automated planning problems is in EXP, and [Rintanen, 2004] shows the case of 2EXP membership under the assumption of partial observability. The membership argument in [Littman, 1997] is by using results on the complexity of solving MDPs [Condon, 1992], showing this to be polynomial in the number of states. As a fully observable planning problem may induce an exponentially sized MDP, this implies the solution existence problem is in EXP. A similar argument is made in [Rintanen, 2004], though instead with reference to the algorithm in [Cimatti et al., 2003].

3.3.4 Non-Branching Problems

Recall that for a non-branching planning problem $\mathcal{P}$, we have for any reachable information cell of $\mathcal{P}$ that any action in the action library is non-branching (see Definition 3.18). So if $\mathcal{P}$ is solvable, then a solution exists which is simply a sequence of actions. However, this observation does not improve the upper bound on SolutionExists, since the number of non-totally bisimilar reachable information cells of $\mathcal{P}$ remains the same as in the case of branching problems — we need another approach. At the same time, it might still be that the length of a solution is (doubly) exponential in the size of $\mathcal{P}$, subject to whether the planning problem is fully or partially observable. Therefore, if the solution existence problem had been a *function problem* (i.e. its output was required to be the strong solution, not just YES or NO), we'd need to store these very large plans and so had no hope of improving the upper bounds shown for SolutionExists.

We now sketch a non-deterministic procedure that decides the solution existence problem for any partially observable and non-branching planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ on P. Starting from $\mathcal{M}_0$, if $\mathcal{M}_0 \models \phi_g$ answer YES, otherwise non-deterministically pick an applicable action $\mathcal{E} \in \mathsf{A}$, and compute the bisimulation contraction of $\mathcal{M} = \mathcal{M}_0 \otimes \mathcal{E}$ which is by definition an information cell. Repeat this process until either $\mathcal{M} \models \phi_g$ in which case the answer is YES, or $2^{2^{|P|}}$ non-deterministic choices have been made in which case the answer is NO.

This procedure is sound, for if the answer is YES then there is a solution to $\mathcal{P}$ consisting of the sequence of actions chosen, each of which are applicable and whose execution is guaranteed to result in an information cell satisfying the goal. It is complete for the following reasons. A solution to this type of planning problem can always be formulated as a sequence of actions, and there are at most $2^{2^{|P|}}$ non-bisimilar information cells on P by Lemma 3.12. Therefore when a solution consists of more than $2^{2^{|P|}}$ actions, it will necessarily reach two bisimilar information cells. Since both applicability and the product update operation are invariant under bisimulation (Corollary 3.7 and Lemma 3.8), this means that for a given solution, removing a sequence of actions between any two bisimilar information cells will also be a solution. Consequently, if $\mathcal{P}$ is solvable then there exists a solution using at most $2^{2^{|P|}}$ actions.

Theorem 3.44. *The solution existence problem for non-branching and partially observable planning problems is in* EXPSPACE.

Proof. Given any partially observable and non-branching planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ on P, and let $i = |\mathcal{P}|$. We show that the procedure outlined above requires no more than $O(2^{i^k})$ space for some constant k. At each step of the computation we need to store a single contraction-minimal information cell, as

well as the number of choices that have been made so far. By Lemma 3.11 we have that a contraction-minimal information cell contains at most $2^{|P|}$ worlds, so as $|P| \leq i$ this requires at most $O(2^i)$ space. As in the proof of Theorem 3.42, we can therefore compute applicability and the product update operation in time polynomial in 2^i, and so this also bounds the amount of space used. Using $2^{|P|}$ bits we can represent any integer up to $2^{2^{|P|}}$, meaning this is the number of bits required to keep of track of how many choices have been made. Therefore the procedure uses only $O(2^{i^k})$ space for some constant k, hence the procedure is in NEXPSPACE and consequently in EXPSPACE by Savitch's theorem. From soundness and completeness of the procedure the result follows. □

In the case of non-branching and fully observable planning problems, we need only to repeat the above procedure until $2^{|P|}$ non-deterministic choices have been made as this is the number of non-bisimilar singleton information cells on P (Lemma 3.10). Storing a singleton information cell requires no more than $O(i)$ space, and using $|P| \leq i$ bits we can keep track of how many choices have been made. From this the following result is established as in the proof above (NPSPACE = PSPACE by Savitch's theorem), except that we must also require actions to be static (preconditions and postconditions belong to $\mathcal{L}_{\mathrm{EL}}(P)$) and the goal formula to belong to $\mathcal{L}_{\mathrm{EL}}(P)$. Otherwise the model checking task may require time exponential in the size of the planning problem, which poses the risk of violating the polynomial space bound. As discussed in Section 3.1.4, if the PSPACE algorithm due [Aucher and Schwarzentruber, 2012] could be modified to handle postconditions, the following result would be voided of these requirements.

Theorem 3.45. *The solution existence problem for non-branching and fully observable planning problems is in* PSPACE*, if each action in the action library is static and the goal formula is in* $\mathcal{L}_{\mathrm{EL}}(P)$*.*

The membership algorithm we sketch here is nondeterministic and uses substantially less space than there are reachable information cells. The key insight upon which this is based is due to [Bylander, 1994], where the solution existence problem for automated planning was shown to be in PSPACE. This insight was also used in [Haslum and Jonsson, 1999] where automated planning problems with partial observability and deterministic actions was shown to be in EXPSPACE. [Bolander and Andersen, 2011] shows that the solution existence problem in the single-agent case is decidable, though the exact complexity is not settled.

As mentioned above, the reason we can obtain better upper bounds is because we're working with a decision problem, and so need not store the entire plan.[6]

[6] Whether we do in fact achieve a better upper bound is subject to whether PSPACE $\stackrel{?}{=}$ EXP and EXPSPACE $\stackrel{?}{=}$ 2EXP.

Nonetheless, this approach is not completely without practical merit when planning and execution is interleaved. Say for instance an agent is given a partially observable and non-branching planning problem that she must solve, and can apply an EXPSPACE algorithm that decides the solution existence problem. By picking an action applicable in the initial state, the agent can query the algorithm and ask whether there exists a strong solution from the resulting information cell. If so the agent can safely execute this action, as it will not render her unable to solve the problem. By repeating this process (for the resulting information cells) until the goal is reached, the agent need not store the entire plan, and so is able to eventually reach a goal state using at most space exponential in the size of the planning problem. This approach can of course be generalized to using a constant number of actions when applying the EXPSPACE algorithm.

3.4 Simulating Turing Machines

In the previous section we gave sound and complete decision procedures for the solution existence problem, and further analyzed their worst-case case complexity. Next on our agenda is to show that each of the four variants of the solution existence problem we consider are among the hardest problems in their indicated complexity class. This means we cannot hope to find better procedures for the solution existence problem than those given above.

The computational complexity of automated planning problems has been extensively studied. [Bylander, 1994] shows the problem of determining whether a solution exists to a classical planning problem is PSPACE-Complete. Adding nondetermism makes the problem EXP-Complete as proved in [Littman, 1997], and with partial observability we have EXPSPACE completeness due to [Haslum and Jonsson, 1999]. Lastly, [Rintanen, 2004] shows that with both partial observability and nondeterminism the problem is 2EXP-Complete. Both [Littman, 1997] and [Haslum and Jonsson, 1999] establish their results by reductions to decision problems known to be complete for their respective classes. The approach of [Bylander, 1994] is more direct in that planning problems are constructed to simulate the transitions of a polynomial space bounded DTM on a given input.

The most generic approach is found [Rintanen, 2004] (and is put on full display in [Rintanen, 2005]), where planning problems are constructed to simulate DTMs and ATMs either with polynomial or exponential space bounds. The two key insights of this work is that nondeterminism can be used to simulate alternating computation, and that partial observability can be used to exponentially increase the number of tape cells that can be represented in a state. What

we present in this section is based on the same ideas, although of course cast in the framework of epistemic planning using our notion of a strong solution. Simulating Turing machines using epistemic planning problems is also done in [Bolander and Andersen, 2011], where it is shown that the solution existence problem is undecidable when using multi-agent DEL and taking plans to be sequences of event models.

We proceed using the approach outlined at the end of Section 3.1.6. More concretely, for each of the four types of planning problems we gave in Section 3.1.5 our modus operandi is as follows.

1. Define the relationship between a given configuration of a TM T and an information cell, namely how the state, head and tape contents are represented as an epistemic model.
2. Define how a transition of T from a configuration c to c' can be simulated by an action, so that the taking the product update of the information cell representing c and said action results is the information cell representing c'.
3. Show that T can make a transition in c exactly when there is an applicable action in the information cell representing c.
4. Construct a planning problem which has a solution exactly when T accepts σ without violating its resource bound.

Two prudent questions may be posed by the reader at this point. Firstly, why do we not give a reduction to the planning problems formulated in [Rintanen, 2004], instead of using similar techniques in showing hardness of our solution existence problem? Secondly, how could things turn out to be any different considering that we're exactly dealing with planning problems that contain nondeterminism and partial observability? We defer our rebuttal to Section 3.5, but invite the impatient reader to look up the discussion before proceeding.

3.4.1 Non-branching and Fully Observable Problems

Theorem 3.46. *The solution existence problem for non-branching and fully observable planning problems is* PSPACE*-Complete.*

Proof. Hardness follows from Proposition 3.50 and Proposition 3.52 (see below), and membership from Theorem 3.45. □

This section is dedicated to showing Theorem 3.46. To this end we consider any deterministic Turing machine $T = (Q, \Sigma, \delta, q_0, l)$ running in polynomial space $p(n)$ on any input σ of length n.

As is our m.o. we start out by defining the relationship between configurations and information cells. Let P be a set of propositional symbols, in which some elements are, conveniently, symbolically identical to their intuitive counterpart in T. We let P be given by

$$\{q \mid q \in Q\} \cup \{h_i \mid 1 \leq i \leq p(n)+1\} \cup$$
$$\{s_i \mid s \in (\Sigma \cup \{\sqcup\}), 1 \leq i \leq p(n)+1\}$$

We call propositional symbols of the type q for *state symbols*, h_i for *head position symbols* and s_i for *letter-position* symbols. The number of symbols in P is therefore $|Q| + (|\Sigma| + 2) \cdot (p(n) + 1)$, and so polynomial in $p(n)$ and the size of T. In what follows we implicitly take models to be defined on P and actions to be members of $\mathcal{L}_{\mathrm{DEL}}(P)$.

Let $c = \sigma_1 q \sigma_2$ be a configuration of T where $|\sigma_1| = n_l$ and $|\sigma_2| = n_r$. We define $\mathcal{M}_c$ as a singleton epistemic model (on P), whose single world satisfies q (the state T is in) and h_{n_l+1} (the position of the head). Furthermore, $\mathcal{M}_c$ satisfies each symbol $\sqcup_i$ where $n_l + n_r < i \leq p(n)$ (unvisited tape cells contain blanks), and finally the singleton epistemic model satisfies each letter-position symbol s_i for which following condition holds.

$$(\sigma_1(i) = s \text{ for } 1 \leq i \leq n_l) \quad \text{or} \quad (\sigma_2(i - n_l) = s \text{ for } n_l < i \leq n_l + n_r)$$

This condition states that each letter-position symbol satisfied by $\mathcal{M}_c$ are those corresponding to the strings σ_1 an σ_2. Note that this definition is unambiguous, because there is no overlap on the conditions of i (nor any overlap with $\sqcup_i$). The result is that $\mathcal{M}_c$ satisfies exactly the symbols corresponding to the state, head position and tape contents as indicated by c (see also Example 3.47). We name $\mathcal{M}_c$ the *information cell associated with* c. For brevity we often skip the denomination and simply write $\mathcal{M}_c$.

Moving on to the correspondence between transitions and actions, consider now some $\delta(q, s) = \{(q', s', d)\}$. For each i such that $1 \leq i \leq p(n)$ we define the action $\mathsf{step}_{i,q,s}^{q',s',d}$ as the singleton event model $(\mathcal{E}, e)$ (of $\mathcal{L}_{\mathrm{DEL}}(P)$) with precondition and postconditions given below (as usual, $post(e)(p) = p$ for those $p \in P$ that are

not specified).

$$
\begin{aligned}
pre(e) &= h_i \wedge q \wedge s_i \\
post(e)(q) &= \top \quad \text{if } q = q' \qquad post(e)(q) = \bot \text{ otherwise} \\
post(e)(q') &= \top \\
post(e)(s_i) &= \top \quad \text{if } s = s' \qquad post(e)(s_i) = \bot \text{ otherwise} \\
post(e)(s_i') &= \top \\
post(e)(h_i) &= \bot \\
post(e)(h_{i-1}) &= \top \quad \text{if } d = L \qquad post(e)(h_{i+1}) = \top \text{ otherwise}
\end{aligned}
$$

Effectively, a single transition of T leads to $p(n)$ actions in the planning problem we construct.

Example 3.47. Figure 3.6 illustrates the relationship between configurations and information cells, as well as that between transitions and actions. On the top left is the configuration $c = xyq_1z$ (we omit blanks when specifying configurations). On the bottom left is the singleton $\mathcal{M}_c$, and displayed below it is a description of the propositional symbols that it satisfies. In this case we have q_1 is satisfied as this is the state of c, and h_3 is satisfied as the machine is reading cell number 3. For each combination of $i \in \{1, \ldots, p(n)\}$ and $s \in (\Sigma \cup \{\sqcup\})$ we have a letter-position symbol, meaning that $x_1, y_2, z_3, \sqcup_4, \ldots, \sqcup_{p(n)}$ all belong to P. More generally, we can describe the tape contents for any configuration (as well as describing information cells not corresponding to configurations).

For the Turing machine in this example we use $\delta(q_1, z) = \{(q_2, z', R)\}$ meaning that c yields $c' = xyz'q_2$. The action simulating this transition is $\mathsf{step}_{3,q_1,z}^{q_2,z',R}$, which we have illustrated at the bottom center of Figure 3.6. Here we list the precondition of the action in the top part. The middle part indicates (using $:=$) the postconditions of individual propositions, namely those propositions whose truth value is changed. The lower part contains a subset of these symbols whose truth value is unchanged. It is clear that this action is applicable in $\mathcal{M}_c$. What is important here is that the head position 3 is fixed for this action (we define $p(n)$ actions for each transition). As we're additionally given the letter to be erased from tape (z), as well as the letter that is to be written (z'), we can form the action so that the postcondition of z_3 is $\bot$ and the postcondition of z_3' is $\top$. The same applies to the head position symbols, where we're given the direction of the transition. For state symbols things are even simpler as these are modified irrespective of the head position. As the truth value of every other propositional symbol is unchanged, the result of the product update is therefore the information cell associated with $\mathcal{M}_{c'}$ indicated at the bottom right.

Lemma 3.48. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $i = |\sigma_1| + 1 \leq p(n)$ being the position of T's head, and where c yields c' due to $\delta(q, s) = \{(q', s', d)\}$.*

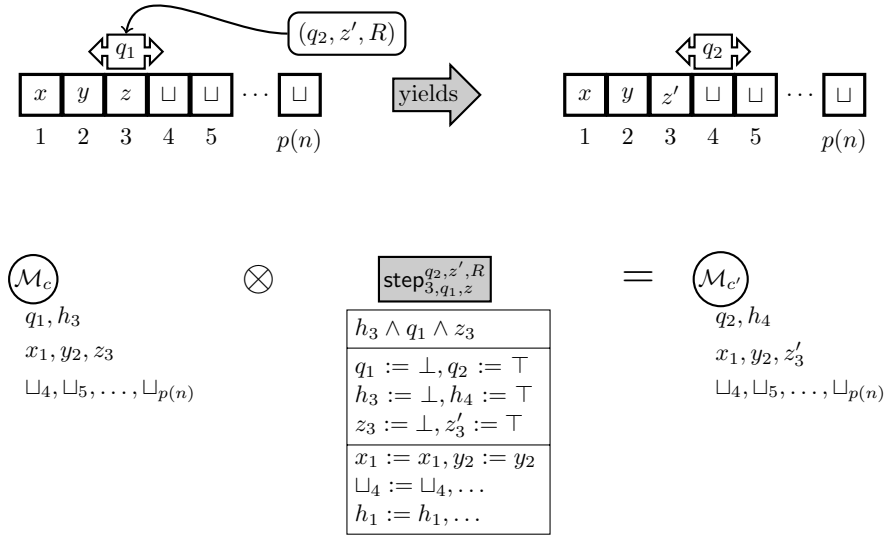


Figure 3.6: Illustration of how information cells are associated with configurations, and the relationship between actions and transitions when simulating a DTM running in polynomial space.

Then $\mathsf{step}^{q',s',d}_{i,q,s}$ *is applicable in* $\mathcal{M}_c$*, and* $\mathcal{M}_c \otimes \mathsf{step}^{q',s',d}_{i,q,s}$ *is the information cell associated with* c'*.*

Proof. See Lemma A.9 in Appendix A.3. □

We now have the ingredients ready for constructing our target planning problem $\mathcal{P}$. The initial state is $\mathcal{M}_{c_0}$, namely the information cell associated with the starting configuration c_0. For each transition $\delta(q, s) = \{(q', s', R)\}$ the action library A contains actions $\mathsf{step}^{q',s',d}_{i,q,s}$ for $1 \leq i \leq p(n)$. Finally, the goal formula is given by

$$\phi_g = \neg h_{p(n)+1} \wedge \bigvee_{\{q \in Q \mid l(q) = \mathsf{acc}\}} q$$

meaning that the space bound must not be violated and that an accepting state is reached. The number of actions is at most $p(n) \cdot |Q| \cdot (|S| + 1)$ and so we have that $\mathcal{P}$ is polynomial in the size of T and $p(n)$. By Proposition 3.19 it follows that $\mathcal{P}$ is non-branching and fully observable.

We now give an auxiliary result. This we use next when we show that if the

starting configuration of T is accepting, then there exists a solution to $\mathcal{P}$; i.e. the "if" part of Theorem 3.46.

Lemma 3.49. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $i = |\sigma_1| + 1 \leq p(n)$, and where c yields c' due to $\delta(q, s) = \{(q', s', d)\}$. Then $\mathcal{M}_c \models \left[\!\left[\mathit{step}_{i,q,s}^{q',s',d} \right]\!\right]_s \phi$ iff $\mathcal{M}_{c'} \models \phi$.*

Proof. Using Lemma 3.48 we have that $\mathsf{step}_{i,q,s}^{q',s',d}$ is applicable in $\mathcal{M}_c$ and so $\mathcal{M}_c \models \langle \mathsf{step}_{i,q,s}^{q',s',d} \rangle \top$. Moreover, $\mathcal{M}_c \otimes \mathsf{step}_{i,q,s}^{q',s',d}$ is isomorphic to the information cell associated with $\mathcal{M}_{c'}$. Therefore $\mathcal{M}_{c'} \models \phi$ iff $\mathcal{M}_c \otimes \mathsf{step}_{i,q,s}^{q',s',d} \models \phi$ iff $\mathcal{M}_c \models [\mathsf{step}_{i,q,s}^{q',s',d}]\phi$ by Lemma 2.16. Recalling Definition 2.11 we conclude $\mathcal{M}_c \models \left[\!\left[\mathsf{step}_{i,q,s}^{q',s',d} \right]\!\right]_s \phi$ iff $\mathcal{M}_{c'} \models \phi$. □

Proposition 3.50. *If the DTM T accepts σ without violating the space bound $p(n)$, then there exists a solution to the non-branching and fully observable planning $\mathcal{P}$.*

Proof. We must show that if c_0 is accepting, then there exists a solution to $\mathcal{P}$. We strengthen this and show that if T is in a configuration $c = \sigma_1 q \sigma_2$ which is x-accepting and $|\sigma_1| + 1 \leq p(n)$, then there is a plan π s.t. $\mathcal{M}_c \models [\![\pi]\!]_s \phi_g$, where $\mathcal{M}_c$ is the information cell associated with c. As $c_0 = q_0 \sigma$ is x-accepting for some $x \in \mathbb{N}$, and the initial state is $\mathcal{M}_{c_0}$, this suffices to prove the result.

We prove this stronger claim by induction on x. If c is 0-accepting then $l(q) = \mathsf{acc}$ and the head position is at most $p(n)$ as T does not violate the space bound. It follows that $\mathcal{M}_c \models q \wedge \neg h_{p(n)+1}$ and therefore $\mathcal{M}_c \models [\![\mathsf{skip}]\!]_s \phi_g$ thus completing the base case. Assume for the induction step that c is $(x+1)$-accepting and $i \leq p(n)$. Then c yields c' due to some $\delta(q, s) = \{(q', s', d)\}$ and c' is y-accepting for some $y \leq x$. To see that the head position of c' is at most $p(n)$, we have from T not violating the space bound that if $i = p(n)$ then $d = L$. From this and c' being y-accepting we can apply the induction hypothesis and conclude that $\mathcal{M}_{c'} \models [\![\pi]\!]_s \phi_g$ for some π. Using Lemma 3.49 it follows that $\mathcal{M}_c \models \left[\!\left[\mathsf{step}_{i,q,s}^{q',s',d} \right]\!\right]_s ([\![\pi]\!]_s \phi_g)$, hence $\mathcal{M}_c \models \left[\!\left[\mathsf{step}_{i,q,s}^{q',s',d}; \pi \right]\!\right]_s \phi_g$ as required. □

We now turn to the "only if" part necessary for Theorem 3.46; i.e. that if a solution exists to $\mathcal{P}$ then the starting configuration of T is accepting. To this end we first show that actions correspond to transitions, that is, if an action is applicable in some information cell associated with a configuration, then the configuration is not halting.

Lemma 3.51. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $i = |\sigma_1| + 1 \leq p(n)$*

and $s = \sigma_2(1)$. If there is an action $\mathcal{E} \in \mathsf{A}$ s.t. such that $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s \phi$, then there is a successor configuration c' of c, and $\mathcal{M}_{c'} \models \phi$.

Proof. For the information cell $\mathcal{M}_c$ associated with c we have that its single world satisfies q, h_i and s_i and no other state, head or letter-position symbols. From $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s \phi$ it follows that $\mathcal{E}$ is applicable in $\mathcal{M}_c$. There is only a single combination in $(\{1, \ldots, p(n)\} \times Q \times S)$ that defines such an applicable action. As $\mathcal{E}$ exists we must have a transition $(q', s', d) \in \delta(q, s)$ and so $\mathcal{E} = \mathsf{step}_{i,q,s}^{q',s',d}$. Therefore c is not halting, and has a successor configuration c'. Using Lemma 3.49 and $\mathcal{M}_c \models \left[\!\left[\mathsf{step}_{i,q,s}^{q',s',d}\right]\!\right]_s \phi$, we conclude $\mathcal{M}_{c'} \models \phi$ as required. □

Proposition 3.52. *If there exists a solution to the non-branching and fully observable planning problem $\mathcal{P}$, then the DTM T accepts σ without violating the space bound $p(n)$.*

Proof. Assume there exists a π for $\mathcal{P}$ s.t. $\mathcal{M}_{c_0} \models [\![\pi]\!]_s \phi_g$. We must show this to imply that $c_0 = q_0\sigma$ is an accepting configuration and T does not violate its space bound. Letting $c = \sigma_1 q \sigma_2$ be a configuration of T s.t. $|\sigma_1| + 1 \leq p(n)$, we will show the stronger result that if $\mathcal{M}_c \models [\![\pi]\!]_s \phi_g$, then c is accepting and no further computation violates the space bound. We proceed by induction on the length of π (cf. Theorem 2.25).

For the base case assume $|\pi| = 1$, meaning that either $\pi = \mathsf{skip}$ or $\pi = \mathcal{E}$ for some $\mathcal{E} \in \mathsf{A}$. The case of $\mathcal{M}_c \models [\![\mathsf{skip}]\!]_s \phi_g$ means that $l(q) = \mathsf{acc}$, hence c is 0-accepting, and as c is a halting configuration no further computation violates the space bound. If $\pi = \mathcal{E}$ then we have $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s \phi_g$ which by Lemma 3.51 means that c yields a configuration c' s.t. $\mathcal{M}_{c'} \models \phi_g$. As in the first case this means c' is 0-accepting and halting, hence c is 1-accepting. As $\mathcal{M}_{c'} \models \neg h_{p(n)+1}$ the postcondition assigned to $h_{p(n)+1}$ for $\mathcal{E}$ is $\bot$. So if $|\sigma_1| + 1 = p(n)$ then this computation moved the head left, and so the space bound has not been violated.

For the induction step assume that $\mathcal{M}_c \models [\![\pi]\!]_s \phi_g$ and $|\pi| = m+1$ for $m \geq 1$. No plan has length 0, and so either $\pi = \mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$, or $\pi = \pi_1; \pi_2$. We recall the two claims shown in the proof of Theorem 2.25, stated here equivalently to suit our needs. First if $\mathcal{M} \models [\![\mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2]\!]_s \phi_g$ then $\mathcal{M} \models [\![\pi_1]\!]_s \phi_g$ or $\mathcal{M} \models [\![\pi_2]\!]_s \phi_g$, with both $|\pi_1| \leq m$ and $|\pi_2| \leq m$ (again, no plan has length 0). Second, if $|\pi_1; \pi_2| \geq 2$ and $\mathcal{M} \models [\![\pi_1; \pi_2]\!]_s \phi_g$ then there exists a plan $\mathcal{E}; \pi'$ s.t. $\mathcal{M} \models [\![\mathcal{E}]\!]_s ([\![\pi']\!]_s \phi_g)$ and $|\pi'| \leq m$.

For the case of $\pi = \mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ we have that $\mathcal{M}_c \models [\![\pi_1]\!]_s \phi_g$ with $|\pi_1| \leq m$ or $\mathcal{M}_c \models [\![\pi_2]\!]_s \phi_g$ with $|\pi_2| \leq m$. Either way we can immediately apply the induction hypothesis, and have that c is accepting and no further computation violates the space bound. In the case where $\pi = \pi_1; \pi_2$, there exists

a π' and some $\mathcal{E}$ s.t. $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s ([\![\pi']\!]_s \phi_g)$ where $|\pi'| \leq m$. By Lemma 3.51 this means that c yields c' s.t. $\mathcal{M}_{c'} \models [\![\pi']\!]_s \phi_g$. To show that the computation from c to c' does not violate the space bound, consider the case of $|\sigma_1|+1 = p(n)$ and assume towards a contradiction that the postcondition of $\mathcal{E}$ assigned to $h_{p(n)+1}$ is $\top$. From this it follows that $\mathcal{M}_{c'} \models h_{p(n)+1}$ and so no action in A is applicable, as actions are not defined for $p(n)+1$. Consequently π' is equivalent to skip, but then follows the falsity $\mathcal{M}_{c'} \models h_{p(n)+1} \wedge ([\![\mathsf{skip}]\!]_s \phi_g)$, hence the postcondition assigned to $h_{p(n)+1}$ is $\bot$. Therefore the head position in c' is at most $p(n)$. From this and $|\pi'| \leq m$ we can apply the induction hypothesis to conclude that c' is x-accepting and no further computation from c' violates the space bound. Therefore c is $(x+1)$-accepting and no further computation from c violates the space bound, thereby completing the induction step. □

3.4.2 Branching and Fully Observable Problems

Theorem 3.53. *The solution existence problem for branching and fully observable planning problems is* EXP*-Complete.*

Proof. Hardness is established from Proposition 3.57 and Proposition 3.60 since APSPACE = EXP. Membership follows from Theorem 3.43. □

To prove the above we tweak the construction of the previous section, which allows us to simulate any *alternating* Turing machine $T = (Q, \Sigma, \delta, q_0, l)$ running in polynomial space $p(n)$ on any input σ of length n. The intuition here is that with actions having multiple basic events and where the epistemic relation is the identity, we can simulate alternation by producing multiple information cells that correspond to configurations. As a strong solution must work for any outcome, this allows us to mimic the behaviour of an ATM in a universal state. The trick of using branching actions for simulating alternation, and the modular manner in which this can be done is due to [Rintanen, 2004].

We can keep P and the notion of an information cell associated with a configuration c as in Section 3.4.1. What we need to change is the actions in the action library, and when doing so we can even use the actions $\mathsf{step}_{i,q,s}^{q',s',d}$ presented above as building blocks. Recall that an existential configuration is accepting if some successor configuration is accepting, and that a universal configuration is accepting if every successor configuration is accepting. Therefore our move to alternating computation requires us to construct actions so that this notion of acceptance is replicated in any solution. For existential configurations we therefore construct one action for each possible transition, whereas universal

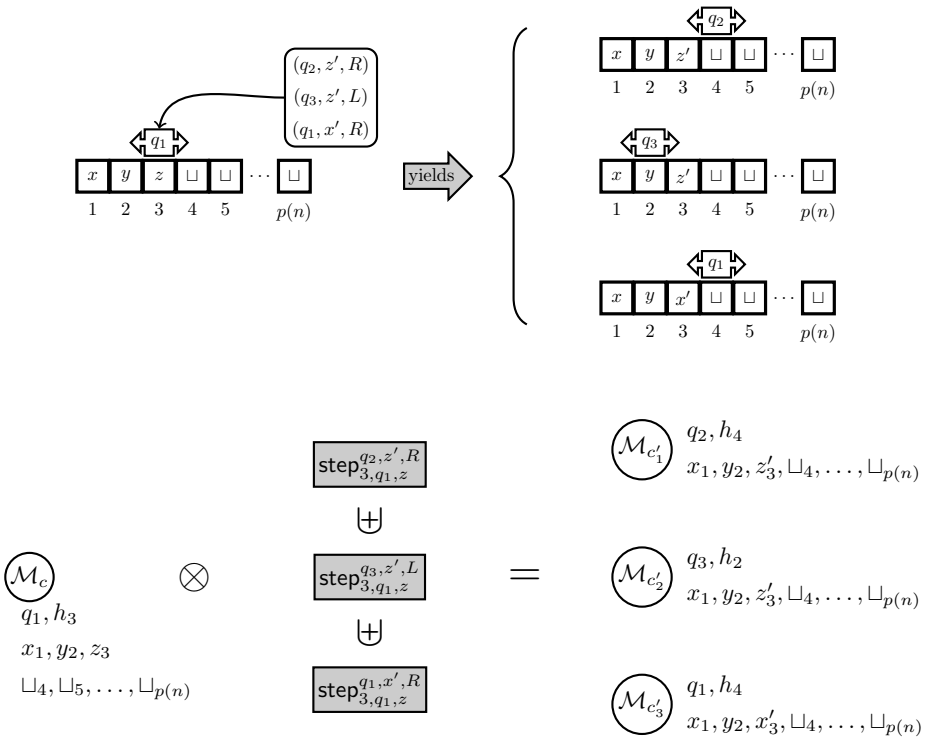


Figure 3.7: Illustration of how a universal transition is associated with an action when simulating an ATM running in polynomial space.

configurations induce only a single action, but this action results in multiple information cells.

Formally, we consider some $\delta(q, s) = \{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$ with i s.t. $1 \leq i \leq p(n)$. If $l(q) = \exists$, then we define k individual actions

$$\mathsf{step}^{q^1,s^1,d^1}_{i,q,s}, \ldots, \mathsf{step}^{q^k,s^k,d^k}_{i,q,s}$$

When instead $l(q) = \forall$, we define a single action

$$\mathsf{step}^{\forall}_{i,q,s} = \mathsf{step}^{q^1,s^1,d^1}_{i,q,s} \uplus \cdots \uplus \mathsf{step}^{q^k,s^k,d^k}_{i,q,s}$$

where we use $\uplus$ to signify the disjoint union of event models. In a fashion similar to Example 3.47, we illustrate how actions simulate universal transitions in Figure 3.7 (preconditions and postconditions of actions are omitted for visual clarity).

Lemma 3.54. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $i = |\sigma_1| + 1 \leq p(n)$ (the head position), and where c has successor configurations $c'_1, \ldots, c'_k$ due to $\delta(q, s) = \{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$.*

- *If $l(q) = \exists$ then for each $j \in \{1, \ldots, k\}$ we have that $\mathsf{step}^{q^j, s^j, d^j}_{i,q,s}$ is applicable in $\mathcal{M}_c$, and $\mathcal{M}_c \otimes \mathsf{step}^{q^j, s^j, d^j}_{i,q,s}$ is the information cell associated with c'_j, and*
- *If $l(q) = \forall$ then $\mathsf{step}^{\forall}_{i,q,s}$ is applicable in $\mathcal{M}_c$, and $\mathcal{M}_c \otimes \mathsf{step}^{\forall}_{i,q,s}$ is the disjoint union of the information cells associated with $c'_1, \ldots, c'_k$.*

Proof. The case of $l(q) = \exists$ is readily established using Lemma 3.48 for each j. For $l(q) = \forall$ recalling the product update operation and noting that the triples in $\{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$ are all pairwise different, the result follows from Lemma 3.48. □

We show some auxiliary results that we use later when proving that if T accepts an input, then there exists a solution to the planning problem. First up is the connection between existential configurations and actions, which is very close to the connection we established in the case of non-branching actions.

Lemma 3.55. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $i = |\sigma_1| + 1 \leq p(n)$, $l(q) = \exists$, and where c has successor configurations $c'_1, \ldots, c'_k$ due to $\delta(q, s) = \{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$. Then for each $j \in \{1, \ldots, k\}$ we have that $\mathcal{M}_c \models \left[\!\left[\mathsf{step}^{q^j, s^j, d^j}_{i,q,s}\right]\!\right]_s \phi$ iff $\mathcal{M}_{c'_j} \models \phi$.*

Proof. As in the proof of Lemma 3.49, here by using Lemma 3.54 rather than Lemma 3.48. □

Connecting transitions for universal states to the plan language requires slightly more elbow grease. For each $(q^j, s^j, d^j) \in \delta(q, s)$ we define $\psi_j = K(h_l \wedge q^j \wedge s^j_l)$, where $l = i - 1$ if $d^j = L$, and $l = i + 1$ if $d^j = R$. We have that ψ_j holds in $\mathcal{M}_{c'_j}$ and that it does not hold in any other information cell in $\mathcal{M}_c \otimes \mathsf{step}^{\forall}_{i,q,s}$, as the triples in $\{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$ are all pairwise different.

Lemma 3.56. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $i = |\sigma_1| + 1 \leq p(n)$, $l(q) = \forall$, and where c has successor configurations $c'_1, \ldots, c'_k$ due to $\delta(q, s) = \{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$. If $\mathcal{M}_{c'_j} \models [\![\pi_j]\!]_s \phi$ for any $j \in \{1, \ldots, k\}$, then $\mathcal{M}_c \models \left[\!\left[\mathsf{step}^{\forall}_{i,q,s}; \mathsf{if}\ \psi_1\ \mathsf{then}\ \pi_1\ \mathsf{else\ if}\ \cdots\ \mathsf{if}\ \psi_k\ \mathsf{then}\ \pi_k\right]\!\right]_s \phi$.*

Proof. By construction of ψ_j we have for any $j \in \{1,\dots,k\}$ that $\mathcal{M}_{c'_j} \models ((\psi_1 \to [\![\pi_1]\!]_s\,\phi) \wedge \dots \wedge (\psi_j \to [\![\pi_j]\!]_s\,\phi)) \wedge (\psi_1 \vee \dots \vee \psi_k)$. With this we can use a derivation like that in the proof of Theorem 2.24 to conclude $\mathcal{M}_{c'_j} \models K\,[\![\mathsf{if}\ \psi_1\ \mathsf{then}\ \pi_1\ \mathsf{else\ if}\ \ \cdots\ \mathsf{if}\ \psi_k\ \mathsf{then}\ \pi_k]\!]_s\,\phi$ (we have $\mathcal{M}_{c'_j} \models \phi \Leftrightarrow \mathcal{M}_{c'_j} \models K\phi$ as $\mathcal{M}_{c'_j}$ is an information cell). Therefore as $\mathcal{M}_c \otimes \mathsf{step}^{\forall}_{i,q,s}$ is the disjoint union of each $\mathcal{M}_{c'_1},\dots,\mathcal{M}_{c'_k}$ (Lemma 3.54), it follows from Lemma 2.16 that

$$\mathcal{M}_c \models \left[\mathsf{step}^{\forall}_{i,q,s}\right](K\,[\![\mathsf{if}\ \psi_1\ \mathsf{then}\ \pi_1\ \mathsf{else\ if}\ \ \cdots\ \mathsf{if}\ \psi_k\ \mathsf{then}\ \pi_k]\!]_s\,\phi)$$

Furthermore $\mathsf{step}^{\forall}_{i,q,s}$ is applicable since $\mathcal{M}_c \models \bigvee_{1\le j\le k} h_i \wedge q \wedge s_i$, so $\mathcal{M}_c \models \langle \mathsf{step}^{\forall}_{i,q,s}\rangle \top$ and cf. Definition 2.11 we have as required:

$$\mathcal{M}_c \models [\![\mathsf{step}^{\forall}_{i,q,s}; \mathsf{if}\ \psi_1\ \mathsf{then}\ \pi_1\ \mathsf{else\ if}\ \ \cdots\ \mathsf{if}\ \psi_k\ \mathsf{then}\ \pi_k]\!]_s\,\phi$$

□

An important note is that ψ_j simply shows the existence of a branching formula — it is not a part of the planning problem given below. If such a formula did not exist the above result could not be established. Constructing our planning problem $\mathcal{P}$ we let the initial state $\mathcal{M}_{c_0}$ and goal formula ϕ_g be as in Section 3.4.1. The action library A contains the actions described above for each $\delta(q,s) = \{(q^1,s^1,d^1),\dots,(q^k,s^k,d^k)\}$ subject to $l(q)$. Both $\mathcal{P}$ and P are polynomial in the size of T and $p(n)$, and Proposition 3.20 implies $\mathcal{P}$ is branching and fully observable.

Proposition 3.57. *If the ATM T accepts σ without violating the space bound $p(n)$, then there exists a solution to the branching and fully observable planning $\mathcal{P}$.*

Proof. We show that if T is in a configuration $c = \sigma_1 q \sigma_2$ which is x-accepting and $|\sigma_1| + 1 \le p(n)$, then there is a plan π s.t. $\mathcal{M}_c \models [\![\pi]\!]_s\,\phi_g$, where $\mathcal{M}_c$ is the information cell associated with c. We proceed by induction on x, with the base being immediate. Assume for the induction step that $c = \sigma_1 q \sigma_2$ is $(x+1)$-accepting, and has successor configurations $c'_1,\dots,c'_k$. If $l(q) = \exists$ then for some $j \in \{1,\dots,k\}$ we have that c'_j is y-accepting for some $y \le x$. The head position of c'_j is at most $p(n)$ as the computation from c to c'_j would otherwise mean T violated the space bound. With this we apply the induction hypothesis, and have that there is a plan π_j s.t. $\mathcal{M}_{c'_j} \models [\![\pi_j]\!]_s\,\phi_g$. Using Lemma 3.55 it follows that $\mathcal{M}_c \models \left[\!\left[\mathsf{step}^{q^j,s^j,d^j}_{i,q,s}\right]\!\right]_s ([\![\pi_j]\!]_s\,\phi_g)$, hence $\mathcal{M}_c \models \left[\!\left[\mathsf{step}^{q^j,s^j,d^j}_{i,q,s}; \pi_j\right]\!\right]_s \phi_g$ as required. If $l(q) = \forall$ then each $c'_1,\dots,c'_k$ is y-accepting for some $y \le x$, and the head position of each such successor configuration is at most $p(n)$. By the induction hypothesis this means there is a plan π_j s.t. $\mathcal{M}_{c'_j} \models [\![\pi_j]\!]_s\,\phi_g$ for each

$j \in \{1, \ldots, k\}$. Using Lemma 3.56 it follows that

$$\mathcal{M}_c \models [\![\mathsf{step}^{\forall}_{i,q,s}; \mathsf{if}\ \psi_1\ \mathsf{then}\ \pi_1\ \mathsf{else\ if}\ \ \cdots\ \mathsf{if}\ \psi_k\ \mathsf{then}\ \pi_k]\!]_s\, \phi_g$$

as required. Assuming that $c_0 = q\sigma$ is an accepting configuration we have that there exists a solution to $\mathcal{P}$. □

With the first direction of Theorem 3.53 being completed, next up is showing that existence of a strong solution is a sufficient condition for the starting configuration to be accepting. The following two results are analogues to Lemma 3.51, now in the context of alternating computation.

Lemma 3.58. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $i = |\sigma_1| + 1 \leq p(n)$ and $l(q) = \exists$. If there is an action $\mathcal{E} \in \mathsf{A}$ such that $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s\, \phi$, then c has a successor configuration c', and $\mathcal{M}_{c'} \models \phi$.*

Proof. For the information cell $\mathcal{M}_c$ associated with c we have that its single world satisfies q, h_i and s_i and no other state, head or letter symbols. From $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s\, \phi$ it follows that $\mathcal{E}$ is applicable in $\mathcal{M}_c$. As $l(q) = \exists$ this means that for the combination (i, q, s) we have defined k actions corresponding to $\delta(q, s) = \{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$, and these are the only actions applicable in $\mathcal{M}_c$. Therefore $\mathcal{E} = \mathsf{step}^{q^j, s^j, d^j}_{i,q,s}$ for some $j \in \{1, \ldots, k\}$, and so c yields c'_j. It follows from Lemma 3.55 and $\mathcal{M}_c \models \left[\!\!\left[\mathsf{step}^{q^j, s^j, d^j}_{i,q,s}\right]\!\!\right]_s \phi$ that $\mathcal{M}_{c'_j} \models \phi$. □

Lemma 3.59. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $i = |\sigma_1| + 1 \leq p(n)$ and $l(q) = \forall$. If there is an action $\mathcal{E} \in \mathsf{A}$ such that $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s\, \phi$, then c has successor configurations $c'_1, \ldots, c'_k$, and $\mathcal{M}_{c'_j} \models \phi$ for each $j \in \{1, \ldots, k\}$.*

Proof. From the same line of reasoning as in the previous proof we can assume that some $\mathsf{step}^{\forall}_{i,q,s}$ is the single applicable action $\mathcal{M}_c$, since $l(q) = \forall$. From Lemma 3.54 it follows that $\mathcal{M}_c \otimes \mathsf{step}^{\forall}_{i,q,s}$ is the disjoint union of information cells associated with $c'_1, \ldots, c'_k$, and since $\mathcal{M}_c \otimes \mathsf{step}^{\forall}_{i,q,s} \models \phi$ we have by Lemma 2.16 that $\mathcal{M}_{c'_j} \models \phi$ for each $j \in \{1, \ldots, k\}$. □

Proposition 3.60. *If there exists a solution to the branching and fully observable planning problem $\mathcal{P}$, then the ATM T accepts σ without violating the space bound $p(n)$.*

Proof. Assuming that there exists a solution π for $\mathcal{P}$ s.t. $\mathcal{M}_{c_0} \models [\![\pi]\!]_s\, \phi_g$, we must show that c_0 is an accepting configuration and T does not violate its space bound. Let $\mathcal{M}_c$ be the information cell associated with some configuration $c = \sigma_1 q \sigma_2$ of T, with $i = |\sigma_1| + 1$ and $i \leq p(n)$. We show the stronger result,

that if $\mathcal{M}_c \models [\![\pi]\!]_s \phi_g$, then c is accepting and no further computation violates the space bound. This is by induction on the length of π.

For the base case $|\pi| = 1$ and we have either $\pi = \mathsf{skip}$ or $\pi = \mathcal{E}$ for some $\mathcal{E} \in \mathsf{A}$. If $\mathcal{M}_c \models [\![\mathsf{skip}]\!]_s \phi_g$ then $\mathcal{M}_c \models \phi_g$, hence $l(q) = \mathsf{acc}$ and c is 0-accepting. If $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s \phi_g$ we have either $l(q) = \exists$ or $l(q) = \forall$ as actions are not defined for halting states. When $l(q) = \exists$ it follows from Lemma 3.58 that c yields some c' s.t. $\mathcal{M}_{c'} \models \phi_g$. Therefore c is 1-accepting as it has a successor configuration c' which is 0-accepting. When $l(q) = \forall$ we apply Lemma 3.59 and have that c has successor configurations $c'_1, \ldots, c'_k$ s.t. $\mathcal{M}_{c'_j} \models \phi_g$ for $j \in \{1, \ldots, k\}$. Therefore each c'_j is 0-accepting and hence c is 1-accepting. As in the proof of Proposition 3.52, any information cell satisfying ϕ_g cannot satisfy $h_{p(n)+1}$, and so the space bound of T is not violated.

For the induction step assume that $|\pi| = m + 1$. As in the proof of Proposition 3.52 we have for $\pi = \mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ that $\mathcal{M}_c \models [\![\pi_1]\!]_s \phi_g$ for $|\pi_1| \leq m$ or that $\mathcal{M}_c \models [\![\pi_2]\!]_s \phi_g$ for $|\pi_2| \leq m$, so applying the induction hypothesis immediately gives that c is accepting and no further computation violates the space bound. Moreover, for $\pi = \pi_1; \pi_2$ we may assume $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s ([\![\pi']\!]_s \phi_g)$ where $|\pi'| \leq m$. If $l(q) = \exists$ it follows from Lemma 3.58 that c has some successor configuration c' with $\mathcal{M}_{c'} \models [\![\pi']\!]_s \phi_g$. As $|\pi'| \leq m$ we can apply the induction hypothesis to conclude that c' is x-accepting for $x \in \mathbb{N}$ and thus c is $(x+1)$-accepting. When $l(q) = \forall$ we have from Lemma 3.59 that c has successor configurations $c'_1, \ldots, c'_k$ s.t. $\mathcal{M}_{c'_j} \models [\![\pi']\!]_s \phi_g$ for each $j \in \{1, \ldots, k\}$. As $|\pi'| \leq m$ we can apply the induction hypothesis and have that each c'_j is x_j-accepting for some $x_j \in \mathbb{N}$, hence c is $(x_{max} + 1)$-accepting where $x_{max} = \max(x_1, \ldots, x_k)$.

Showing that no further computation from c violates the space bound is for $l(q) = \exists$ exactly as in the proof of Proposition 3.52, viz. using proof by contradiction. The same applies to $l(q) = \forall$, where by assuming that the head position of any c'_j is such that $\mathcal{M}_{c'_j} \models h_{p(n)+1}$ we derive the contradiction $\mathcal{M}_{c'_j} \models h_{p(n)+1} \wedge ([\![\mathsf{skip}]\!]_s \phi_g)$. □

3.4.3 Non-branching and Partially Observable Problems

Next up are planning problems with partial observability, and for this we turn to Turing machines running in exponential space. As noted previously, when we later make the move from DTMs to ATMs, we can reuse the above approach for simulating alternation. First up however, is showing how configurations with exponentially many letters on the tape can be represented.

We let $T = (Q, \Sigma, \delta, q_0, l)$ be a deterministic Turing machine running in exponential space $e(n)$ on any input σ of length n. Adding partial observability allows us to use actions whose epistemic relation is not the identity. In fact we only need such actions for some initial steps; actions for simulating transitions are *still* singletons. In effect we now have non-singleton information cells at our disposal, and this allows us to encode an entire configuration, including the exponentially in n sized tape, as a single information cell. In the presentation that follows, we do not rely on the notions developed in Section 3.4.1 and Section 3.4.2, even though we for convenience use some overlapping notation. The result we're after first is this.

Theorem 3.61. *The solution existence problem for non-branching and partially observable planning problems is* EXPSPACE*-Complete.*

Proof. Hardness follows from Proposition 3.66 and Proposition 3.67. Membership follows from Theorem 3.44. □

We'll be using two ideas that we credit [Rintanen, 2004]. First is the use of propositional symbols as bit encodings of the head position, and further giving each world a distinct identifier, which allows succinct actions to faithfully simulate transitions (Rintanen does not use an identifier per say, rather he keeps track of a watched tape cell randomly chosen at the start of plan execution). Here our construction is very similar. Second is the use of partial observability to represent exponentially many possibilities. Here our construction differs, because our model of observation is unlike that of [Rintanen, 2004], and so we do not have the option of succinctly specifying an initial state that contains exponentially many possibilities. We overcome this by employing a bootstrapping procedure.

The first component of the simulation is to show how configurations and information cells correspond. We will use $b = \lceil \log_2(e(n) + 1) \rceil$, which is enough bits to represent any integer between 1 and $e(n) + 1$. Note that b is polynomial in n. We let P be the following set of symbols.

$$\{q \mid q \in Q\} \cup \{s \mid s \in (\Sigma \cup \{\sqcup\})\} \cup$$
$$\{h_i \mid 1 \leq i \leq b\} \cup \{id_i \mid 1 \leq i \leq b\} \cup$$
$$\{init_i \mid 0 \leq i \leq b\}$$

We say that propositional symbols of the type q is a *state symbol*, s is a *letter symbol*, h_i is a *head position bit*, id_i is an *identification bit* and $init_i$ is an *initialization control symbol* (we purposely introduce more control symbols than we do bits).

The first thing to note is that unlike the case of polynomial space TMs, we

do not form letter-position symbols for representing tape contents. Instead, we will be giving each world in an information cell a unique identifier (the role of identification bits), which represents its position on tape. Moreover, we also encode the current position of the head using the head position bits, which changes as the head moves. We use $init_i$ for bootstrapping the initial state of the planning problem, a procedure which we return to later. For convenience we write h to indicate the integer represented by $h_1, \ldots, h_b$ and similarly with id for $id_1, \ldots id_b$. Moreover, for any integer j between 1 and 2^b, we use $j_1, \ldots, j_b$ to denote its bit (0's and 1's) representation.

We'll now formally show what the information cells we're after look like (see also Figure 3.8). Consider a configuration $c = \sigma_1 q \sigma_2$ of T where $|\sigma_1| = n_l$ and $|\sigma_2| = n_r$. We define $\mathcal{M}_c = (W, \sim, V)$ on P as the information cell associated with c. Here $W = \{w_j \mid 1 \leq j \leq 2^b\}$ and $\sim$ is (by definition) the universal relation on W; i.e. our information cells contain 2^b worlds. We give the valuation V so that state symbols and head identification bits have the same truth value in every world, as these are global properties of c. Further, each world is given a distinct id to indicate a cell index as well as the symbol corresponding to the letter at this particular cell index. For each world $w_j \in W$ this means:

$$
\begin{array}{lll}
w_j \in V(id_i) & \text{if } j_i \text{ is } 1 & j \text{ is interpreted as an integer} \\
w_j \in V(h_i) & \text{if } (n_l + 1)_i \text{ is } 1 & n_l + 1 \text{ is the position of the head} \\
V(q) = W & & \\
V(q') = \emptyset \text{ for } q' \in (Q \setminus \{q\}) & &
\end{array}
$$

$$
w_j \in V(s) \text{ if } \begin{cases} j \leq n_l \text{ and } \sigma_1(j) = s \\ j > n_l, j \leq n_l + n_r \text{ and } \sigma_2(j - n_l) = s \\ j > n_l + n_r, j \leq e(n) + 1 \text{ and } s = \sqcup \end{cases} \quad \text{for } s \in (\Sigma \cup \{\sqcup\})
$$

$$
V(init_b) = W
$$

The three cases for $w_j \in V(s)$ imply that w_j is at most in one valuation of $s \in (\Sigma \cup \{\sqcup\})$, and that if $j > e(n) + 1$ we associate it with no symbol. We have defined $\mathcal{M}_c$ so that it contains least one more world than there are tape cells, which means that space bound violation is well-defined (detection follows from the goal formula). As a technical quirk $V(init_b) = W$ is used to indicate that bootstrapping has taken place.

Example 3.62. We illustrate an information cell associated with some configuration $c = \sigma_1 q \sigma_2$ of T in Figure 3.8. We'll let $|\sigma_1| + 1 = m$ be the position of the head, and assume $\sigma_2(1) = x$ is currently read. This means $\mathcal{M}_c$ contains a world w_m whose id and h bit encodings are both equal to m. In accordance with the discussion below this example, we simply write $id = m$ and $h = m$ to indicate this. Also, w_m satisfies the state symbol q as well as $init_b$, and finally it satisfies x. Say that x is also written on the cell of some other position o.

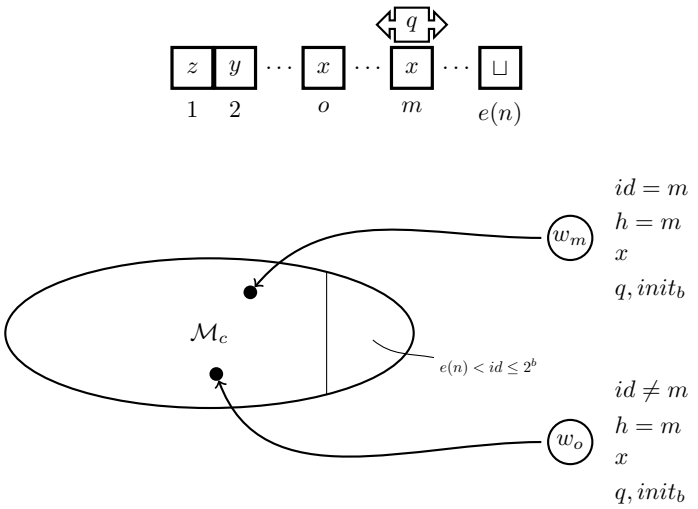


Figure 3.8: Illustration of an information cell associated with a configuration, zooming in on two worlds. The information cell contains 2^b worlds.

This means we have a world w_o where still $h = m$, but instead $id \neq m$ (since $o \neq m$). As was the case for w_m, we have that w_o satisfies q, $init_b$ and x. In total $\mathcal{M}_c$ contains 2^b worlds, each with a distinct id. Moreover, worlds for which $id > e(n) + 1$ are assigned no letter symbol.

We will be employing bit comparison and bit manipulation in the remainder. While nothing more than boolean operations, we include how this is encoded for the sake of completeness. For integers j, j' we first introduce some typical relations. We write $j = j'$ to abbreviate $(j_1 \leftrightarrow j'_1 \wedge \cdots \wedge j_b \leftrightarrow j'_b)$, $j \neq j'$ to abbreviate $\neg(j = j')$, $j > j'$ to abbreviate $((j_b \wedge \neg j'_b) \vee ((j_b \leftrightarrow j'_b) \wedge (j_{b-1} \wedge \neg j'_{b-1})) \vee \cdots \vee (((j_b \leftrightarrow j'_b) \wedge \cdots \wedge (j_2 \leftrightarrow j'_2)) \wedge (j_1 \wedge \neg j'_1)))$, and finally $j \leq j'$ to abbreviate $((j = j') \vee \neg(j > j'))$. These formulas are polynomial in the size of b. We also use formulas to encode integer incrementation and decrementation. For $1 \leq i \leq b$ we write $post(e)(j) = j{+}1$ to mean $post(e)(j_i) = j_i \leftrightarrow \neg(j_1 \wedge \cdots \wedge j_{i-1})$ (the empty conjunction is as usual equivalent to $\top$). We do so similarly for $post(e)(j) = j - 1$. As we noted at the end of Section 3.2, these operations can be used to exponentially increase the number of nodes in our Fibonacci construction.

If the initial configuration of T is $c_0 = q_0\sigma$, then $\mathcal{M}_{c_0}$ will be exponential in the size of T. To avoid this we use a bootstrapping procedure, starting from a singleton information cell $\mathcal{M}_s$ satisfying exactly $init_0$. $\mathcal{M}_s$ is "blown up" so that it contains 2^b worlds, and then each world is assigned a letter symbol in

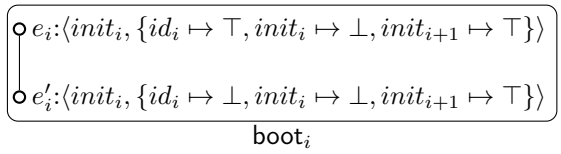


Figure 3.9: Action used for bootstrapping the planning problem.

accordance with the input σ. The procedure is controlled by $init_i$, first using the actions boot_i for $0 \leq i < b$ given in Figure 3.9. Then using the action $\mathsf{finalize}$ which is a singleton event model $(\mathcal{E}, e_f)$, where:

$$
\begin{aligned}
pre(e_f) &= init_b \wedge \bigwedge_{q \in Q} \neg q \\
post(e_f)(q_0) &= \top \\
post(e_f)(\sqcup) &= (n < id) \wedge (id \leq e(n) + 1) \\
\forall s \in \Sigma : post(e_f)(s) &= \bigvee_{\substack{\{j \mid \sigma(j) = s, \\ 1 \leq j \leq n\}}} (id = j)
\end{aligned}
$$

With this we have a method for transforming a singleton initial state into the information cell associated with the starting configuration, which is formally stated as follows.

Lemma 3.63. *If* $\mathcal{M}_{c_0} \models \phi$, *then* $\mathcal{M}_s \models [\![\mathsf{boot}_0; \ldots; \mathsf{boot}_{b-1}; \mathsf{finalize}]\!]_s \phi$.

Proof. $\mathcal{M}_s$ satisfies $init_0$, and as each boot_i results in $init_{i+1}$ being true in every world these actions are sequentially applicable. $\mathsf{finalize}$ is applicable as boot_{b-1} results in $init_b$ being satisfied and no $q \in Q$ holds. The information cell $\mathcal{M}_s \otimes \mathsf{boot}_0 \otimes \cdots \otimes \mathsf{boot}_{b-1}$ contains 2^b worlds, each having a distinct id between 1 and 2^b. Therefore $\mathsf{finalize}$ assigns truth in accordance with c_0, and so $\mathcal{M}_s \otimes \mathsf{boot}_0 \otimes \cdots \otimes \mathsf{boot}_{b-1} \otimes \mathsf{finalize}$ is exactly the information cell associated with $\mathcal{M}_{c_0}$. From this and Lemma 2.16 we can readily derive the conclusion. □

Importantly, none of the bootstrapping actions are applicable in an information cell associated with a configuration of T, so this is a true initialization procedure.

Turning now to actions for representing transitions, consider some $\delta(q, s) = \{(q', s', d)\}$. We define the action $\mathsf{estep}^{q',s',d}_{q,s}$ as the singleton event model $(\mathcal{E}, e)$

with:

$$
\begin{array}{lll}
 & pre(e) = init_b \wedge q \wedge (h \leq e(n)) \wedge \widehat{K}(s \wedge (h = id)) & \\
\text{if } q = q' & post(e)(q) = \top & \text{o.w. } post(e)(q) = \bot \\
 & post(e)(q') = \top & \\
\text{if } d = L & post(e)(h) = h - 1 & \text{o.w. } post(e)(h) = h + 1 \\
\text{if } s = s' & post(e)(s) = s & \text{o.w. } post(e)(s) = (h \neq id) \wedge s \\
\text{if } s = s' & post(e)(s') = s' & \text{o.w. } post(e)(s') = (h = id) \vee (h \neq id \wedge s')
\end{array}
$$

Consider now some information cell $\mathcal{M}$ on P. Intuitively $pre(e)$ means that $\mathsf{estep}_{q,s}^{q',s',d}$ is applicable in $\mathcal{M}$ when $\mathcal{M}$ has been bootstrapped, q holds in every world of $\mathcal{M}$, the head is within the allotted part of the tape, and finally that there *is some* world satisfying s and whose id equals the head's position. Stating the latter condition as e.g. $(s \wedge (id = h)) \vee (id \neq h))$ would allow changing state and moving the head without updating any letter (when $s \neq s'$), and so the use of the dual knowledge modality is vital here.

Example 3.64. Figure 3.10 illustrates the manner in which we simulate a transition. On the top left we have some configuration $c = \sigma_1 q_1 \sigma_2$ of T with $|\sigma_1| + 1 = m$ and $\sigma_2(1) = x$. We then consider that $\delta(q_1, x) = \{(q_2, z', L)\}$ whose corresponding action is $\mathsf{estep}_{q_1,x}^{q_2,z',L}$. In the top part of the rectangle below $\mathsf{estep}_{q_1,x}^{q_2,z',L}$ we show its precondition. The precondition need not mention m, rather using the dual modality it requires some world to satisfy $x \wedge (h = id)$. By construction we have that every world of $\mathcal{M}_c$ satisfies $h = m$, and additionally that each world is assigned a distinct id. Therefore exactly one world w_m satisfies $h = m = id$.

The middle part of the rectangle below $\mathsf{estep}_{q_1,x}^{q_2,z',L}$ shows the postconditions of propositions whose truth value may be modified. The bottom part mentions a subset of the symbols whose truth value is unchanged. Here we update q_1 and q_2 to reflect that this transition changes the state of T, and additionally h is decremented by 1 to signify a move to the left. With $x := (h \neq id) \wedge x$ we state that the action preserves the truth value of x in every world of $\mathcal{M}_c$, *except* those in which $h = id$ where x becomes false. In other words, x is set false for w_m and its truth value is otherwise retained. Using $z' := (h = id) \vee (h \neq id \wedge z')$ we similarly achieve that z' is set true for w_m, and the right-hand side of the disjunct states that z' retains its truth value in every other world.

Analogous to Lemma 3.48 we have the following.

Lemma 3.65. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $|\sigma_1| + 1 \leq e(n)$, that yields c' due to $\delta(q, s) = \{(q', s', d)\}$. Then $\mathsf{estep}_{q,s}^{q',s',d}$ is applicable in $\mathcal{M}_c$, and $\mathcal{M}_c \otimes \mathsf{estep}_{q,s}^{q',s',d}$ is the information cell associated with c'.*

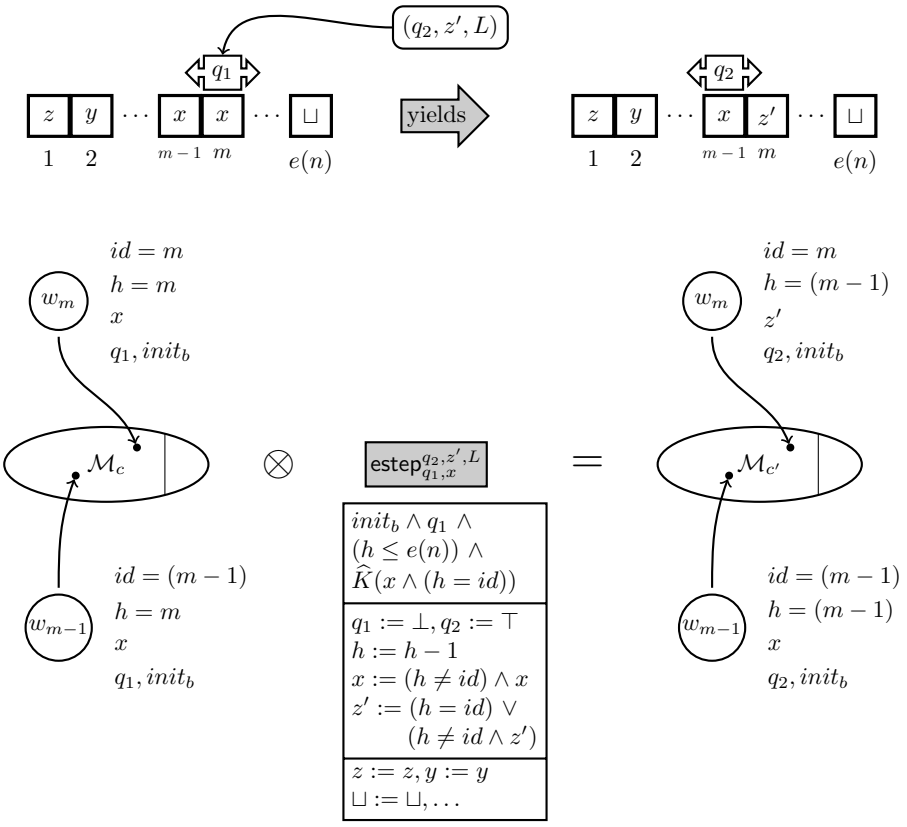


Figure 3.10: Simulating a transition for a DTM running in exponential space.

Proof. See Lemma A.10 in Appendix A.3. □

Using all of the above we're now ready to give our planning problem $\mathcal{P}$ for simulating T. The initial state is $\mathcal{M}_s$, that is, the singleton information cell used at the start of bootstrapping satisfying exactly $init_0$. The action library contains the bootstrapping actions $\mathsf{boot}_0, \ldots, \mathsf{boot}_{b-1}$, $\mathsf{finalize}$, and additionally an action $\mathsf{estep}^{q',s',d}_{q,s}$ for each transition $\delta(q,s) = \{(q',s',d)\}$. Finally the goal formula is

$$\phi_g = \bigvee_{\{q \in Q \mid l(q) = \mathsf{acc}\}} q \wedge (h \leq e(n))$$

The planning problem and P are polynomial in the size of T and $\log(e(n))$, and Proposition 3.21 implies $\mathcal{P}$ is non-branching and partially observable.

Proposition 3.66. *If the DTM T accepts σ without violating the space bound*

$e(n)$, then there exists a solution to the non-branching and partially observable planning problem $\mathcal{P}$.

Proof. The proof is along the lines of that used when T runs in polynomial space, namely by induction x, showing that for any configuration c that is x-accepting, there exists a strong solution for $\mathcal{M}_c$. The addition here is the use of Lemma 3.63 for additionally showing that any such plan can be prefixed with $\mathsf{boot}_0; \ldots; \mathsf{boot}_{b-1}; \mathsf{finalize}$ to work from the initial state $\mathcal{M}_c$. The result is fully shown in the proof of Lemma A.12 in Appendix A.3. □

For the other direction of Theorem 3.61 we have the following.

Proposition 3.67. *If there exists a solution to the non-branching and partially observable planning problem $\mathcal{P}$, then the DTM T accepts σ without violating the space bound $e(n)$.*

Proof. First we can show that the bootstrapping procedure must always take place in a strong solution to $\mathcal{P}$. Following this we proceed along the lines of the proof of Proposition 3.52, that is, by induction on the length of a solution for any information cell associated with a configuration. Again the goal formula ensures the space bound of T is not violated if there exists a solution to $\mathcal{P}$, and further that c_0 is an accepting configuration. For a full proof see Lemma A.14 in Appendix A.3. □

3.4.4 Branching and Partially Observable Problems

Last on our agenda in the following result.

Theorem 3.68. *The solution existence problem for branching and partially observable planning problems is* 2EXP*-Complete.*

Proof. Hardness is established from Proposition 3.70 and Proposition 3.71 since AEXPSPACE = 2EXP. Membership follows from Theorem 3.42. □

We can augment the constructions of the previous section so that alternating computation can be simulated. Doing so is decidedly close to the techniques of Section 3.4.2, where we lifted DTMs with a polynomial space bound to ATMs with same bound. Therefore we shall only briefly describe the construction, and place full proofs in the appendix.

Without further ado, we let $T = (Q, \Sigma, \delta, q_0, l)$ be any alternating Turing machine running in exponential space $e(n)$ on any input σ of length n. We use the same set of symbols P and same notion of information cell associated a configuration as in Section 3.4.3. Furthermore, the actions presented in said section are building blocks for the actions used to to simulate the transitions $(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)$ in $\delta(q, s)$. When $l(q) = \exists$, then we define k individual actions, viz. $\mathsf{estep}^{q^1,s^1,d^1}_{q,s}, \ldots, \mathsf{estep}^{q^k,s^k,d^k}_{q,s}$. If instead $l(q) = \forall$ we define a single action $\mathsf{estep}^{\forall}_{q,s} = \mathsf{estep}^{q^1,s^1,d^1}_{q,s} \uplus \cdots \uplus \mathsf{estep}^{q^k,s^k,d^k}_{q,s}$.

Having read Section 3.4 up until this point, the only novelty is the existence of branching formulas. Just as when we presented Lemma 3.56, we require the existence of such formulas for solution to be able to distinguish the outcomes of an $\mathsf{estep}^{\forall}_{q,s}$ action. To this end define for each $(q^j, s^j, d^j) \in \delta(q, s)$ the formula

$$\psi_j = K(\widehat{K}((id = h_l) \wedge q^j \wedge s^j))$$

where $h_l = h - 1$ if $d^j = L$, and $h_l = h + 1$ if $d^j = R$, and we note that h_l is compared bitwise to id. We have that ψ_j holds in $\mathcal{M}_{c'_j}$ and that it does not hold in any other information cell in $\mathcal{M}_c \otimes \mathsf{estep}^{\forall}_{q,s}$, due to the fact that the triples in $\{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$ are all pairwise different. As was the case when we gave preconditions for our actions, the dual modality $\widehat{K}$ is important for defining ψ_j, because it dictates the *existence* of a world satisfying $(id = h_l) \wedge q^j \wedge s^j$. We stress that ψ_j constructively shows the existence of a branching formula for distinguishing the information cells of $\mathcal{M}_c \otimes \mathsf{estep}^{\forall}_{i,q,s}$ — it is not part of the reduction.

Lemma 3.69. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $|\sigma_1| + 1 \leq e(n)$, $l(q) = \forall$, and where c has successor configurations $c'_1, \ldots, c'_k$ due to $\delta(q, s) = \{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$. If $\mathcal{M}_{c'_j} \models [\![\pi_j]\!]_s \phi$ for any $j \in \{1, \ldots, k\}$, then $\mathcal{M}_c \models [\![\mathsf{estep}^{\forall}_{i,q,s}; \mathsf{if}\ \psi_1\ \mathsf{then}\ \pi_1\ \mathsf{else\ if}\ \cdots\ \mathsf{if}\ \psi_k\ \mathsf{then}\ \pi_k]\!]_s \phi$.*

Proof. See the proof of Lemma A.17 in Appendix A.3. □

The planning problem $\mathcal{P}$ we construct is as in Section 3.4.3 except for the action library, meaning the initial state is $\mathcal{M}_s$ (satisfies exactly $init_0$) and the goal is $\phi_g = \bigvee_{\{q \in Q \mid l(q) = \mathsf{acc}\}} q \wedge (h \leq e(n))$. The action library A contains the bootstrapping actions $\mathsf{boot}_0, \ldots, \mathsf{boot}_{b-1}$, $\mathsf{finalize}$, and additionally for each $\delta(q, s) = \{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$, the k actions $\mathsf{estep}^{q^1,s^1,d^1}_{q,s}, \ldots, \mathsf{estep}^{q^k,s^k,d^k}_{q,s}$ if $l(q) = \exists$, or the action $\mathsf{estep}^{\forall}_{q,s}$ if $l(q) = \forall$. We have that $\mathcal{P}$ and P are polynomial in the size of T and $\log(e(n))$.

Proposition 3.70. *If the ATM T accepts σ without violating the space bound $e(n)$, then there exists a solution to the branching and partially observable planning problem $\mathcal{P}$.*

Proof. See the proof of Lemma A.18 in Appendix A.3. □

Proposition 3.71. *If there exists a solution to the branching and partially observable planning problem $\mathcal{P}$, then the ATM T accepts σ without violating the space bound $e(n)$.*

Proof. See the proof of Lemma A.21 in Appendix A.3. □

3.5 Conclusion and Discussion

In this chapter we took at closer look at the computational complexity of conditional epistemic planning as propagated in Chapter 2, namely by investigating the decision problem of whether a strong solution exists to a planning problem. Our first step was to illustrate that using the StrongPlan algorithm for this purpose was infeasible. We then took the natural step and introduced planning graphs as an alternative to planning trees, subsequently showing that the notion of a solved planning tree corresponds to that of a solved planning graph. Our results were with regards to four types of planning problems, separated by the level of observability (partial or full) and whether or not actions were branching. We gave the procedure SolutionExists for optimally deciding the solution existence problem in the case of branching planning problems. For non-branching planning problems, where a solution can always be found that is sequential, we could improve the upper bound of the corresponding solution existence problem. Having shown membership, we proceeded to construct epistemic planning problems that simulated Turing machines, and used this to show that each of the four variants of the solution existence problem were complete for their respective complexity class. The results are summarized in Figure 3.11, and we note that along the way we used insights that must be credited [Bylander, 1994, Rintanen, 2004].

We now return to the two prudent questions mentioned at the beginning of Section 3.4. Our sincere answer as to why we have not presented a reduction from the plan existence problem of [Rintanen, 2004] to our solution existence problem here is simply that we have not been able to find one that works in the general case. Because the complexity classes we deal with are all closed under polynomial-time reductions, we have however established that a reduction exist in both directions. The two reductions that would be interesting and natural would allow for transforming operators (as given in [Rintanen, 2004, Definition 2]) to event models and event models to operators.

From operators to event models the issue is that an operator may be very

Partial Observability	Branching	Complexity Class
No	No	PSPACE (Theorem 3.46)
No	Yes	EXP (Theorem 3.53)
Yes	No	EXPSPACE (Theorem 3.61)
Yes	Yes	2EXP (Theorem 3.68)

Figure 3.11: Summary of complexity results for the solution existence problem; right-most column indicates the problem is complete for the given class. The result for full observability and non-branching planning problems assumes static actions and that the goal is a member of $\mathcal{L}_{\mathrm{EL}}(P)$.

succinct, that is, the effect of one operator may produce exponentially many states. We could likely overcome this in a natural way, by adding sequential composition and nondeterministic union to $\mathcal{L}_{\mathrm{DEL}}(P)$ (cf. [van Ditmarsch et al., 2007]), which would allow us the same succinctness of representation when using event models. From event models to operators the problems are more dire. We model what can be observed by virtue of the information cells that result from a product update operation. In this sense each such information cell corresponds to some kind of observation. In Rintanen's framework, partial observability is handled by letting branching formulas be propositional formulas over a given set of observation variables [Rintanen, 2004, Definition 5]. We therefore need to find an operator and a set of observation variables, so that following the application of said operator, a branching formula can precisely distinguish the outcomes that do not belong to the same information cell. This we have not been able to do.

The ways in which our simulation of Turing machines differs from Rintanen's provides some insights into the differences of the two frameworks. The succinctness of operators is superfluous for simulating transitions, and so we do not run into trouble when using event models for this purpose. On the other hand, the move to partial observability requires us to employ a bootstrapping procedure, in order for information cells to encode exponentially sized tapes. Rintanen achieves this by giving an initial state formula that effectively describes exponentially many states.

The second question was how our results could have been any different, considering we're dealing solely with planning for a single agent. What we have shown is that single-agent conditional epistemic planning *does* capture the spectrum of automated planning (complexity wise), ranging from classical planning to nondeterministic planning with partial observability. We see this as a feature of the conditional epistemic planning framework, confirming that event models adequately capture operators, at least in terms of the types of planning problems

they can describe (cf. discussion of succinctness above). That this need not be the case is exemplified in the "KBPs as Plans" approach of [Lang and Zanuttini, 2013]. Here an action cannot be both fully observable and branching, meaning that the class of planning problems that are EXP-Complete is not captured.

Using the results in this chapter we can conclude that, for better or for worse, the computational complexity associated with conditional epistemic planning coincides with the results known from automated planning. In terms of asymptotic complexity, we need not be put off by our framework and surrender to the Markovian route usually taken in automated planning. From a methodological perspective we have illustrated that epistemic model theory is crucial in developing optimal decision procedures for the solution existence problem. While the results from epistemic model theory that we have used are not particularly sophisticated, they provide us with just what we need. We can keep the size of an information cell down by computing its bisimulation contraction, and we can put an upper bound on the number of distinct information cells we may reach. In going beyond single-agent epistemic planning, notions from model theory are likely to play an equally important role. Incidentally, it is exactly the model theory of epistemic-plausibility models that we take to investigate in the ensuing chapter.

Chapter 4

Bisimulation for single-agent plausibility models

This chapter is a replicate of [Andersen et al., 2013] which appers in the proceedings of the 26th Australasian Conference on Artificial Intelligence, 2013, in Otago, New Zealand. In the following presentation a few minor corrections have been made.

Bisimulation for single-agent plausibility models

Mikkel Birkegaard Andersen* Thomas Bolander*
Hans van Ditmarsch† Martin Holm Jensen*
*DTU Compute, Technical University of Denmark
†LORIA, CNRS, Université de Lorraine

Abstract

Epistemic plausibility models are Kripke models agents use to reason about the knowledge and beliefs of themselves and each other. Restricting ourselves to the single-agent case, we determine when such models are indistinguishable in the logical language containing conditional belief, i.e., we define a proper notion of bisimulation, and prove that bisimulation corresponds to logical equivalence on image-finite models. We relate our results to other epistemic notions, such as safe belief and degrees of belief. Our results imply that there are only finitely many non-bisimilar single-agent epistemic plausibility models on a finite set of propositions. This gives decidability for single-agent epistemic plausibility planning.

4.1 Introduction

A typical approach in belief revision involves preferential orders to express degrees of belief and knowledge [Kraus et al., 1990, Meyer et al., 2000]. This goes back to the 'systems of spheres' in [Lewis, 1973, Grove, 1988]. Dynamic doxastic logic was proposed and investigated in [Segerberg, 1998] in order to provide a link between the (non-modal logical) belief revision and modal logics with

explicit knowledge and belief operators. A similar approach was pursued in belief revision in dynamic epistemic logic [Aucher, 2005, van Ditmarsch, 2005, van Benthem, 2007, Baltag and Smets, 2008, van Ditmarsch and Labuschagne, 2007], that continues to develop strongly [Britz and Varzinczak, 2013, van Benthem, 2011]. We focus on the proper notion of structural equivalence on (static) models encoding knowledge and belief simultaneously. A prior investigation into that is [Demey, 2011], which we relate our results to at the end of the paper. Our motivation is to find suitable structural notions to reduce the complexity of planning problems. Such plans are sequences of actions, such as iterated belief revision. It is the dynamics of knowledge and belief that, after all, motivates our research.

The semantics of belief depend on the structural properties of models. To relate the structural properties of models to a logical language we need a notion of structural similarity, known as bisimulation. A bisimulation relation relates a modal operator to an accessibility relation. Epistemic plausibility models do not have an accessibility relation as such but a plausibility relation. This induces a set of accessibility relations: the *most plausible* states are the *accessible* states for the modal belief operator; and the *plausible* states are the *accessible* states for the modal knowledge operator. But it contains much more information: to each modal operator of conditional belief (or of degree of belief) one can associate a possibly distinct accessibility relation. This begs the question how one should represent the bisimulation conditions succinctly. Can this be done by reference to the plausibility relation directly, instead of by reference to these, possibly many, induced accessibility relations? It is now rather interesting to observe that relative to the modal operations of knowledge and belief the plausibility relation is already in some way too rich.

Example 4.1. The (single-agent) epistemic plausibility model on the left in Figure 4.1 consists of three worlds w_1, w_2, and w_3. p is only false in w_2, and $w_1 < w_2 < w_3$[1]: the agent finds it most plausible that p is true, less plausible that p is false, and even less plausible that p is true. As p is true in the most plausible world, the agent believes p. If we go to slightly less plausible, the agent is already uncertain about the value of p, she only knows trivialities such as $p \vee \neg p$. The world w_3 does not make the agent even more uncertain. We therefore can discard that other world where p is true. This is the model in the middle in Figure 4.1. It is bisimilar to the model on the left! Therefore, and that is the important observation: *having one world more or less plausible than another world in a plausibility model does not mean that in any model with the same logical content we should find a matching pair of worlds.* This is evidenced in the figure: on the left w_3 is less plausible than w_2, but in the middle no world is less plausible than v_2; there is no match.

[1] If $s < t$, we have $s \leq t$ and $t \not\leq s$.

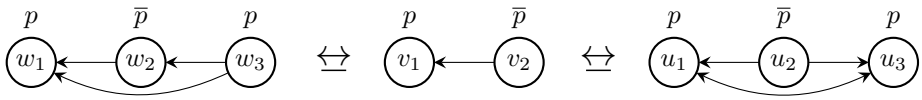


Figure 4.1: All three models are bisimilar. The models in the middle and on the right are normal, the model on the left is not normal. An arrow $w_1 \leftarrow w_2$ corresponds to $w_1 \leq w_2$. Reflexive edges are omitted. $\overline{p}$ means that p does not hold.

Now consider retaining w_3 and making it as plausible state as w_1. This gives the plausibility model on the right in Figure 4.1, where u_1 and u_3 are equiplausible (equally plausible), written $u_1 \simeq u_3$. This model is bisimilar to both the left and the middle model. But the right and middle one share the property that more or less plausible in one, is more or less plausible in the other: now there is a match. This makes for another important observation: *we can reshuffle the plausibilities such that models with the same logical content preserve the plausibility order.*

In Section 4.2 we define the epistemic doxastic logic, the epistemic plausibility models on which it is interpreted, the suitable notion of bisimulation, and demonstrate the adequacy of this notion via a correspondence between modal equivalence and bisimilarity. The final sections 4.3, 4.4, and 4.5 respectively translate our results to degrees of belief and safe belief, discuss the problematic generalization to the multi-agent case, and demonstrate the relevance of our results for epistemic planning.

4.2 Single-agent plausibility models and bisimulation

4.2.1 Language, structures, and semantics

Definition 4.2 (Epistemic doxastic language)**.** For any countable set of propositional symbols P, we define the epistemic-doxastic language $\mathcal{L}_P$ by:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K\varphi \mid B^{\varphi}\varphi$$

where $p \in P$, K is the epistemic modality (knowledge) and B^{φ} the conditional doxastic modality (conditional belief). We use the usual abbreviations for the other boolean connectives as well as for $\top$ and $\bot$, and the abbreviation B for $B^{\top}$. The dual of K is denoted $\widehat{K}$, and the dual of B^{φ} is denoted $\widehat{B}^{\varphi}$.

We consider epistemic plausibility models as in [Baltag and Smets, 2008]. A *well-preorder* on a set S is a reflexive and transitive relation $\leq$ on S such that every non-empty subset has minimal elements. The set of *minimal elements* of a subset T of S is given by:

$$Min_{\leq}T = \{s \in T \mid s \leq s' \text{ for all } s' \in T\}.$$

This is a non-standard notion of minimality, taken from [Baltag and Smets, 2008]. Usually a minimal element of a set is an element that is not greater than any other element. On total preorders the two notions of minimality coincide. In fact, using the definition of minimality above, any well-preorder is total: For any pair of worlds s, t, $Min_{\leq}\{s, t\}$ is non-empty, and therefore $s \leq t$ or $t \leq s$.[2] These well-preorders are the *plausibility relations* (or *plausibility orderings*), expressing that a world is considered at least as plausible as another. This encodes the doxastic content of a model.

We can define such epistemic plausibility models with the plausibility relation as a primitive and with the epistemic relation as a derived notion. Alternatively, we can assume both as primitive relations, but require that more plausible means (epistemically) possible. We chose the latter.

Definition 4.3 (Epistemic plausibility model)**.** An *epistemic plausibility model* (or simply *plausibility model*) on a set of propositional symbols P is a tuple $\mathcal{M} = (W, \leq, \sim, V)$, where

- W is a set of *worlds*, called the *domain*.
- $\leq$ is a well-preorder on W, called the *plausibility relation*.
- $\sim$ is an equivalence relation on W called the *epistemic relation*. We require, for all $w, v \in W$, that $w \leq v$ implies $w \sim v$.
- $V : W \to 2^P$ is a *valuation*.

For $w \in W$ we name $(\mathcal{M}, w)$ a *pointed epistemic plausibility model*, and refer to w as the *actual world* of $(\mathcal{M}, w)$.

As we require that $\leq$-comparable worlds are indistinguishable, totality of $\leq$ gives that $\sim$ is the universal relation $W \times W$.

[2] A well-preorder is not the same as a well-founded preorder; e.g., $y \leq x$, $z \leq x$ is a well-founded preorder, but not a well-preorder, as z and y are incomparable. Well-founded preorders are not necessarily total.

Definition 4.4 (Satisfaction Relation)**.** Let $\mathcal{M} = (W, \leq, \sim, V)$ be a plausibility model on P. The satisfaction relation is given by, for $w \in W$, $p \in P$, $\varphi, \varphi' \in \mathcal{L}_P$,

$$\begin{array}{lll} \mathcal{M}, w \models p & \text{iff} & p \in V(w) \\ \mathcal{M}, w \models \neg\varphi & \text{iff} & \text{not } \mathcal{M}, w \models \varphi \\ \mathcal{M}, w \models \varphi \wedge \varphi' & \text{iff} & \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \varphi' \\ \mathcal{M}, w \models K\varphi & \text{iff} & \mathcal{M}, v \models \varphi \text{ for all } v \sim w \\ \mathcal{M}, w \models B^{\psi}\varphi & \text{iff} & \mathcal{M}, v \models \varphi \text{ for all } v \in Min_{\leq}[\![\psi]\!]_{\mathcal{M}}, \end{array}$$

where $[\![\psi]\!]_{\mathcal{M}} := \{w \in W \mid \mathcal{M}, w \models \psi\}$. We write $\mathcal{M} \models \varphi$ to mean $\mathcal{M}, w \models \varphi$ for all $w \in W$. Further, $\models \varphi$ (φ is valid) means that $\mathcal{M} \models \varphi$ for all models $\mathcal{M}$, and $\Phi \models \varphi$ (φ is a logical consequence of the set of formulas Φ) stands for: for all $\mathcal{M}$ and $w \in \mathcal{M}$, if $\mathcal{M}, w \models \psi$ for all $\psi \in \Phi$, then $\mathcal{M}, w \models \varphi$.[3]

Example 4.5. Consider again the the models in Figure 4.1. The model on the left is of the form $\mathcal{M} = (W, \leq, \sim, V)$ with $W = \{w_1, w_2, w_3\}$ and $\leq$ defined by: $w_1 \leq w_2$, $w_2 \leq w_3$, $w_1 \leq w_3$ (plus the reflexive edges). The valuation V of the model on the left maps w_1 into $\{p\}$, w_2 into $\emptyset$ and w_3 into $\{p\}$. In all three models of the figure, the formula $Bp \wedge \neg Kp$ holds, that is, p is believed but not known.

4.2.2 Normal epistemic plausibility models and bisimulation

The examples and proposal of Section 4.1 are captured by the definition of bisimulation that follows after these preliminaries. First, given a plausibility model $\mathcal{M} = (W, \sim, \leq, V)$ consider an equivalence relation on worlds defined as follows:

$$w \approx w' \quad \text{iff} \quad V(w) = V(w').$$

The $\approx$-equivalence class of a world is defined as usual as $[w]_{\approx} = \{w' \in W \mid w' \approx w\}$. Next, the ordering $\leq$ on worlds in W can be lifted to an ordering between sets of worlds $W', W'' \subseteq W$ in the following way:

$$W' \leq W'' \quad \text{iff} \quad w' \leq w'' \text{ for all } (w', w'') \in W' \times W''.$$

Finally, the lifted ordering leads us to a formalization of normal models of Example 4.1.

Definition 4.6 (Normal Plausibility Relation)**.** Given a plausibility model $\mathcal{M} = (W, \leq, \sim, V)$, the *normal plausibility relation* on $\mathcal{M}$ is the relation on W defined by:

$$w \preceq w' \quad \text{iff} \quad Min_{\leq}[w]_{\approx} \leq Min_{\leq}[w']_{\approx}.$$

[3] For an axiomatization of this logic see e.g. [Stalnaker, 1996].

$\mathcal{M}$ is called *normal* if $\preceq = \leq$. The *normalisation* of $\mathcal{M} = (W, \leq, \sim, V)$ is $\mathcal{M}' = (W, \preceq, \sim, V)$. As for $<$, we write $w \prec w'$ for $w \preceq w'$ and $w' \not\preceq w$.

Note that if $u, v \in Min_{\leq} W'$ for some set W' then, by definition of $Min_{\leq}$, both $u \leq v$ and $v \leq u$. Hence, the condition $Min_{\leq}[w]_{\approx} \leq Min_{\leq}[w']_{\approx}$ above is equivalent to the existence of *some* minimal element of $[w]_{\approx}$ being $\leq$-smaller than *some* minimal element of $[w']_{\approx}$.

Lemma 4.7. *Let w and w' be two worlds in the normal model $\mathcal{M} = (W, \preceq, \sim, V)$. If w and w' have the same valuation, they are equiplausible.*

Proof. As $w \approx w'$, we have $[w]_{\approx} = [w']_{\approx}$, and thus $Min_{\preceq}[w]_{\approx} = Min_{\preceq}[w']_{\approx}$. By Definition 4.6 we have $w \preceq w'$ and $w' \preceq w$, which is equivalent to $w \simeq w'$. □

Example 4.8. Take another look at the models of Figure 4.1 (for reference, we name them $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$). We want models $\mathcal{M}_1$ and $\mathcal{M}_2$ to be bisimilar via the relation $\mathfrak{R}$ given by $\mathfrak{R} = \{(w_1, v_1), (w_3, v_1), (w_2, v_2)\}$ (see Section 4.1). Usually, in a bisimulation, every modal operator has corresponding back and forth requirements. For our logic of conditional belief there is an infinity of modal operators, as there is an infinity of of conditional formulas. (Having *only* unconditional belief $B\varphi$ defined as $B^{\top}\varphi$ is not enough, see Example 4.13.) Instead, we define our bisimulation indirectly by way of the plausibility relation. Example 4.1 showed that we cannot match 'more plausible' in $\mathcal{M}_1$ with 'more plausible' in $\mathcal{M}_2$ using simply $\leq$. With $\leq$ as seen in $\mathcal{M}_3$ (the normalization of $\mathcal{M}_1$) where $\leq = \preceq$, we can.

Definition 4.9 (Bisimulation)**.** Let plausibility models $\mathcal{M} = (W, \leq, \sim, V)$ and $\mathcal{M}' = (W', \leq', \sim', V')$ be given. Let $\preceq, \preceq'$ be the respective derived normal plausibility relations. A non-empty relation $\mathfrak{R} \subseteq W \times W'$ is a *bisimulation* between $\mathcal{M}$ and $\mathcal{M}'$ if for all $(w, w') \in \mathfrak{R}$:

[atoms] $V(w) = V'(w')$.

[forth$_{\preceq}$] If $v \in W$ and $v \preceq w$, there is a $v' \in W'$ s.t. $v' \preceq' w'$ and $(v, v') \in \mathfrak{R}$.

[back$_{\preceq}$] If $v' \in W'$ and $v' \preceq' w'$, there is a $v \in W$ s.t. $v \preceq w$ and $(v, v') \in \mathfrak{R}$.

[forth$_{\sim}$] If $v \in W$ and $w \sim v$, there is a $v' \in W'$ s.t. $w' \sim' v'$ and $(v, v') \in \mathfrak{R}$.

[back$_{\sim}$] If $v' \in W'$ and $w' \sim' v'$, there is a $v \in W$ s.t. $w \sim v$ and $(v, v') \in \mathfrak{R}$.

A *total bisimulation* between $\mathcal{M}$ and $\mathcal{M}'$ is a bisimulation with domain W and codomain W'. For a bisimulation between pointed models $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$

it is required that $(w, w') \in \mathfrak{R}$. If a bisimulation between $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ exists, the two models are called *bisimilar* and we write $(\mathcal{M}, w) \leftrightarrows (\mathcal{M}', w')$. Two worlds w, w' of a model $\mathcal{M}$ are called *bisimilar* if there exists a bisimulation $\mathfrak{R}$ between $\mathcal{M}$ and itself with $(w, w') \in \mathfrak{R}$.

This definition gives us the bisimulation put forth in Example 4.8. As $\sim$ is the universal relation on W, [forth$_\sim$] and [back$_\sim$] enforce that all bisimulations are total.

If $\sim$ was not a primitive, we could instead have conditions [up-forth$_\preceq$] and [up-back$_\preceq$] (that consider less plausible v and v'), in place of [forth$_\sim$] and [back$_\sim$]. This would define the same bisimulations.

4.2.3 Correspondence between bisimilarity and modal equivalence

In the following we prove that bisimilarity implies modal equivalence and vice versa. This shows that our notion of bisimulation is proper for the language and models at hand. First we define modal equivalence.

Definition 4.10 (Modal equivalence)**.** Given are models $\mathcal{M} = (W, \leq, \sim, V)$ and $\mathcal{M}' = (W', \leq', \sim', V')$ on P with $w \in W$ and $w' \in W'$. We say that $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ are *modally equivalent* iff for all $\varphi \in \mathcal{L}_P$, $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}', w' \models \varphi$. In this case we write $(\mathcal{M}, w) \equiv (\mathcal{M}', w')$.

Lemma 4.11. *If two worlds of a model are $\approx$-equivalent, they are bisimilar.*

Proof. Assume worlds w and w' of a model $\mathcal{M} = (W, \leq, \sim, V)$ have the same valuation. Let $\mathfrak{R}$ be the relation that relates each world of $\mathcal{M}$ to itself and additionally relates w to w'. We want to show that $\mathfrak{R}$ is a bisimulation. This amounts to showing [atoms], [forth$_\preceq$], [back$_\preceq$], [forth$_\sim$] and [back$_\sim$] for the pair $(w, w') \in \mathfrak{R}$. [atoms] holds trivially since $w \approx w'$. [forth$_\sim$] and [back$_\sim$] also hold trivially, by choice of $\mathfrak{R}$. For [forth$_\preceq$], assume $v \in W$ and $v \preceq w$. We need to find a $v' \in W$ such that $v' \preceq w'$ and $(v, v') \in \mathfrak{R}$. Letting $v' = v$, it suffices to prove $v \preceq w'$. Since $w \approx w'$ this is immediate: $v \preceq w$ iff $Min_\leq[v]_\approx \leq Min_\leq[w]_\approx$ iff (because $w \approx w'$) $Min_\leq[v]_\approx \leq Min_\leq[w']_\approx$ iff $v \preceq w'$. [back$_\preceq$] is proved similarly. □

Proposition 4.12. *Bisimilarity implies modal equivalence.*

Proof. We will prove that for all formulas $\varphi \in \mathcal{L}_P$, if $\mathfrak{R}$ is a bisimulation between pointed models $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ then $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}', w' \models \varphi$. The proof is by induction on the structure of φ. The base case is when φ is propositional. Then the required follows immediately from [atoms], using that $(w, w') \in \mathfrak{R}$. For the induction step, we have the following cases of φ: $\neg\psi, \psi \wedge \gamma, K\psi, B^\gamma\psi$. We skip the first three, fairly standard cases and show only $B^\gamma\psi$.

Let $\mathfrak{R}$ be a bisimulation between $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ with $\mathcal{M} = (W, \leq, \sim, V)$ and $\mathcal{M} = (W', \leq', \sim', V')$. We only prove $\mathcal{M}, w \models B^\gamma\psi \Rightarrow \mathcal{M}', w' \models B^\gamma\psi$, the other direction being proved symmetrically. So assume $\mathcal{M}, w \models B^\gamma\psi$, that is, $\mathcal{M}, v \models \psi$ for all $v \in \mathit{Min}_{\leq}[\![\gamma]\!]_{\mathcal{M}}$. We need to prove $\mathcal{M}', v' \models \psi$ for all $v' \in \mathit{Min}_{\leq'}[\![\gamma]\!]_{\mathcal{M}'}$. So let $v' \in \mathit{Min}_{\leq'}[\![\gamma]\!]_{\mathcal{M}'}$. Choose $x \in \mathit{Min}_{\leq}\{u \in W \mid u \approx z \text{ and } (z, v') \in \mathfrak{R} \text{ for some } z \in W\}$. Let $y \in [\![\gamma]\!]_{\mathcal{M}}$ be chosen arbitrarily, and choose y' with $(y, y') \in \mathfrak{R}$ (recall that any bisimulation is total). The induction hypothesis implies $\mathcal{M}', y' \models \gamma$. Let $y'' \approx y'$ be chosen arbitrarily. Lemma 4.11 implies the existence of a bisimulation $\mathfrak{R}'$ between $(\mathcal{M}', y'')$ and $(\mathcal{M}', y')$. Since $\mathcal{M}', y' \models \gamma$, the induction hypothesis gives us $\mathcal{M}', y'' \models \gamma$, that is, $y'' \in [\![\gamma]\!]_{\mathcal{M}'}$. Since v' was chosen $\leq'$-minimal in $[\![\gamma]\!]_{\mathcal{M}'}$, we must have $v' \leq' y''$. Since y'' was chosen arbitrarily with $y'' \approx y'$, we get $v' \leq' \mathit{Min}_{\leq'}[y']_{\approx}$. We can now conclude $\mathit{Min}_{\leq'}[v']_{\approx} \leq' v' \leq' \mathit{Min}_{\leq'}[y']_{\approx}$, and hence $v' \preceq y'$.

By [back$_\preceq$] there is a v such that $(v, v') \in \mathfrak{R}$ and $v \preceq y$. By choice of x, $x \leq \mathit{Min}_{\leq}[v]_{\approx}$. Since $v \preceq y$ we now get: $x \leq \mathit{Min}_{\leq}[v]_{\approx} \leq \mathit{Min}_{\leq}[y]_{\approx} \leq y$. Since y was chosen arbitrarily in $[\![\gamma]\!]_{\mathcal{M}}$, we can conclude:

$$x \leq u \text{ for all } u \in [\![\gamma]\!]_{\mathcal{M}}. \tag{4.1}$$

By choice of x, there is a $z \approx x$ with $(z, v') \in \mathfrak{R}$. From $z \approx x$, Lemma 4.11 implies the existence of a bisimulation $\mathfrak{R}''$ between $(\mathcal{M}, x)$ and $(\mathcal{M}, z)$. Since $\mathfrak{R}''$ is a bisimulation between $(\mathcal{M}, x)$ and $(\mathcal{M}, z)$, and $\mathfrak{R}$ is a bisimulation between $(\mathcal{M}, z)$ and $(\mathcal{M}', v')$, the composition $\mathfrak{R}'' \circ \mathfrak{R}$ must be a bisimulation between $(\mathcal{M}, x)$ and $(\mathcal{M}', v')$. Applying the induction hypothesis to the bisimulation $\mathfrak{R}'' \circ \mathfrak{R}$, we can from $v' \in [\![\gamma]\!]_{\mathcal{M}'}$ conclude $x \in [\![\gamma]\!]_{\mathcal{M}}$. Combining this with (4.1), we get $x \in \mathit{Min}_{\leq}[\![\gamma]\!]_{\mathcal{M}}$. By original assumption this implies $\mathcal{M}, x \models \psi$. Applying again the induction hypothesis to the bisimulation $\mathfrak{R}'' \circ \mathfrak{R}$, this gives us $\mathcal{M}, v' \models \psi$, as required, thereby concluding the proof. □

We proceed now to the converse, that modal equivalence with regard to $\mathcal{L}_P$ implies bisimulation, though first taking a short detour motivating the need for conditional belief.

Example 4.13. The normal plausibility models $(\mathcal{M}_1, w_1)$ and $(\mathcal{M}_2, v_1)$ of Figure 4.2 are modally equivalent for the language with only unconditional belief. We can show this by first demonstrating that $\mathcal{M}_1$ and $\mathcal{M}_2$ have the same modal

description Φ (a modal description Φ of a model $\mathcal{M}$ is a set of formulas such that $\Phi \models \psi$ iff $\mathcal{M} \models \psi$). We observe that the description of both models is

$$B(p_1 \wedge \neg p_2 \wedge \neg p_3) \wedge K((p_1 \wedge \neg p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge p_3))$$

To see why, note that w_1 and v_1 are both the only minimal worlds in their respective models, so belief in (description of the valuation) $p_1 \wedge \neg p_2 \wedge \neg p_3$ will be the same. Further, in both models all three constituent worlds are epistemically possible, so K cannot distinguish either between the models (the disjunction sums up the three different valuations). We then note that, as both w_1 and v_1 satisfy $p_1 \wedge \neg p_2 \wedge \neg p_3$, $(\mathcal{M}_1, w_1)$ and $(\mathcal{M}_2, v_1)$ of Figure 4.2 must be modally equivalent: any boolean formula must be a consequence of $p_1 \wedge \neg p_2 \wedge \neg p_3$, whereas any belief or knowledge formula evaluated in the points of these models must be a model validity that is a consequence from the model description Φ.

On the other hand, $(\mathcal{M}_1, w_1)$ and $(\mathcal{M}_2, v_1)$ are not bisimilar. Pairs in the bisimulation must have matching valuations, so the only option is the relation $\{(w_1, v_1), (w_2, v_3), (w_3, v_2)\}$. But this does neither satisfy [forth$_\preceq$] nor [back$_\preceq$].

We do not want that these models are modally equivalent in, for example, a *dynamic* epistemic language. Consider an agent learning $\neg p_1$ from a public announcement. This deletes w_1 and v_1 from their respective models. After this announcement in $\mathcal{M}_1$, the agent believes p_2. In $\mathcal{M}_2$ this is not the case. Here the agent will believe p_3. With conditional belief we can capture this distinction already in the static language ($\mathcal{M}_1 \models B^{\neg p_1} p_2$, while $\mathcal{M}_2 \not\models B^{\neg p_1} p_2$).

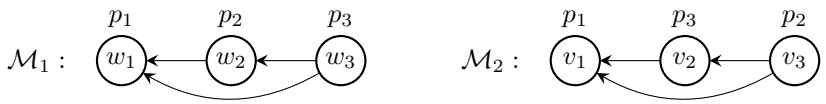


Figure 4.2: The models $\mathcal{M}_1$ and $\mathcal{M}_2$ of Example 4.13. For visual clarity, we leave out false propositional variables.

Definition 4.14 (Δ)**.** Let two worlds w, w' of a model $\mathcal{M} = (W, \leq, \sim, V)$ on P be given where $V(w) \neq V(w')$. If there is a $p \in V(w) - V(w')$, then let $\delta_{w,w'}$ be such a p; otherwise, let $\delta_{w,w'} = \neg q$ for some $q \in V(w') - V(w)$. Any such choice of $\delta_{w,w'}$ for a given pair w, w' is called a *propositional difference* between w and w'. If instead $V(w) = V(w')$, let $\delta_{w,w'} = \top$. Finally, let $\Delta_w = \bigwedge_{w' \prec w} \delta_{w,w'}$ be the conjunction of some propositional difference between w and each world strictly more $\preceq$-plausible than w (the empty conjunction when no such world exist).

Continuing Example 4.13, we can choose $\Delta_{w_2} = \neg p_1$. We then have that $\widehat{B}^{\Delta_{w_2}} p_2$ distinguishes $\mathcal{M}_1$ and $\mathcal{M}_2$ by evaluating belief on worlds no more plausible than

w_2 and v_2 respectively. However, choosing $\Delta_{w_2} = p_2$ would not distinguish, so we add an additional disjunct for w_3. Regardless of which propositional differences are used in Δ_{w_2} and Δ_{w_3}, $\widehat{B}^{\Delta_{w_2} \vee \Delta_{w_3}} p_2$ distinguishes the models. This is, of course, not sufficient for constructing distinguishing formulas in the general case, but for our purposes of proving Proposition 4.16 it is enough.

Lemma 4.15. *Let w and w' be worlds of the model $\mathcal{M} = (W, \leq, \sim, V)$ s.t. $w' \preceq w$, and φ a formula of $\mathcal{L}_P$, s.t. $\mathcal{M}, w' \models \varphi$. Then $\mathcal{M}, w \models \widehat{B}^{\Delta_w \vee \Delta_{w'}} \varphi$.*

Proof. In the following we abbreviate $\Delta_w \vee \Delta_{w'}$ by $\Delta_{w,w'}$. We need to show that $\exists u \in Min_{\leq}[\![\Delta_{w,w'}]\!]_{\mathcal{M}}$, s.t. $\mathcal{M}, u \models \varphi$. By construction of $\Delta_{w,w'}$, we have that for all $s \in [\![\Delta_{w,w'}]\!]_{\mathcal{M}}$, either $s \approx w$, $s \approx w'$ or ($w \prec s$ and $w' \prec s$). By choice of w and w', we have $w' \preceq w$, meaning that $\exists w'' \in Min_{\leq}[\![\Delta_{w,w'}]\!]_{\mathcal{M}}$ such that $w' \approx w''$. Lemma 4.11 then says that w' and w'' are bisimilar, and Proposition 4.12 that they are modally equivalent. Thus $\mathcal{M}, w'' \models \varphi$. This is the u we are looking for, giving $\mathcal{M}, w \models \widehat{B}^{\Delta_{w,w'}} \varphi$.

□

Proposition 4.16. *On the class of image-finite models, modal equivalence implies bisimilarity.*

Proof. Let $\mathcal{M} = (W, \leq, \sim, V)$ and $\mathcal{M}' = (W', \leq', \sim', V')$ be two image-finite, plausibility models on P, and define $\mathfrak{R} \subseteq W \times W'$, such that $(w, w') \in \mathfrak{R}$ iff $(\mathcal{M}, w) \equiv (\mathcal{M}', w')$. We show that $\mathfrak{R}$ is in fact a bisimulation of the kind defined in Definition 4.9. Showing that $\mathfrak{R}$ satisfies [atoms] is trivial. We skip the, less trivial, [forth$_\sim$], and [back$_\sim$] and show the considerably more complicated case of [forth$_\preceq$] ([back$_\preceq$] is similar) as follows: Assume $(\mathcal{M}, w) \equiv (\mathcal{M}', w')$, $v \in W$ and $v \preceq w$ and show that assuming that for all $v' \in W'$, $v' \preceq w'$ implies $(\mathcal{M}, v) \not\equiv (\mathcal{M}', v')$, leads to a contradiction. This then gives $(\mathcal{M}, v) \equiv (\mathcal{M}', v')$ and therefore $(v, v') \in \mathfrak{R}$.

Let $S' = \{v' \mid v' \preceq w'\} = \{v'_1, \ldots v'_n\}$ be the successors of w'. This set is finite, due to image-finiteness of the model. If v and no successor of w' is modally equivalent, there exists formulae $\varphi^{v'_i}$, such that $\mathcal{M}, v \models \varphi^{v'_i}$ and $\mathcal{M}', v'_i \not\models \varphi^{v'_i}$. Therefore, $\mathcal{M}, v \models \varphi^{v'_1} \wedge \cdots \wedge \varphi^{v'_n}$. For notational ease, let $\Phi = \varphi^{v'_1} \wedge \cdots \wedge \varphi^{v'_n}$.

With $\mathcal{M}, v \models \Phi$, Lemma 4.15 gives $\mathcal{M}, w \models \widehat{B}^{\Delta_{w,v}} \Phi$ ($\Delta_{w,v}$ is finite due to image-finiteness of the models). Now, $\mathcal{M}', w' \models \widehat{B}^{\Delta_{w,v}} \Phi$ (which we must have due to modal equivalence) iff there exists a $u' \in Min_{\leq}[\![\Delta_{w,v}]\!]_{\mathcal{M}'}$ such that $\mathcal{M}', u' \models \Phi$. By construction of Φ, no world v'_i exists such that $v'_i \preceq w'$ and $\mathcal{M}', v'_i \models \Phi$, so we must have $w' \prec u'$. There are two cases for (the weakest requirements for)

this u' to be minimal. Either (i) $u' \leq w'$ or (ii) $w' < u'$ and $w' \notin [\![\Delta_{w,v}]\!]_{\mathcal{M}'}$. If (i) is the case, we must have a world w'', with $w'' \approx w'$ and $w'' < u'$, or we couldn't have $w' \prec u'$. But $w'' < u'$ means that u' cannot be minimal unless $w' \notin [\![\Delta_{w,v}]\!]_{\mathcal{M}'}$, because otherwise $w'' \in [\![\Delta_{w,v}]\!]_{\mathcal{M}'}$. So, for (i) and (ii) both, we must have $w' \notin [\![\Delta_{w,v}]\!]_{\mathcal{M}'}$. This yields $\mathcal{M}', w' \models \neg\Delta_{w,v}$. But as $\mathcal{M}, w \models \Delta_{w,v}$, we get the sought after contradiction of $(\mathcal{M}, w) \equiv (\mathcal{M}', w')$.

□

4.3 Degrees of belief and safe belief

In this section we sketch some further results that can be obtained for our single-agent setting of the logic of knowledge and conditional belief. Apart from *conditional belief*, other familiar epistemic notions in the philosophical logical and artificial intelligence community are *safe belief* [Stalnaker, 1996] and *degrees of belief* [Kraus et al., 1990, Spohn, 1988]. Our results generalize fairly straightforwardly to such other notions. An agent has *safe belief* in formula φ iff it will continue to believe φ no matter what *true* information conditions its belief.[4]

Definition 4.17 (Safe belief)**.** We extend the inductive language definition with a clause $\Box\varphi$ for safe belief in φ. The semantics are $\mathcal{M}, w \models \Box\varphi$ iff $(\mathcal{M}, w \models B^{\psi}\varphi$ for all ψ such that $\mathcal{M}, w \models \psi)$.

Degrees of belief are a quantitative alternative to conditional belief. The zeroth degree of belief $B^0\varphi$ is defeasible belief $B\varphi$ as already defined. For $\mathcal{M}, w \models B^1\varphi$ to hold φ should be true in (i) all minimal worlds accessible from w; but additionally, (ii) if you take away those from the equivalence class, in all worlds that are now minimal. If we do this with the normal plausibility relation we get what we want (otherwise, we run into the same problems as before — our treatment is not compatible with e.g. Spohn's approach [Spohn, 1988], that allows 'gaps' (layers without worlds) in between different degrees of belief).

$$
\begin{array}{lcll}
Min^0_{\preceq}[w]_{\sim} & := & Min_{\preceq}([w]_{\sim}) & \\
Min^{\overline{n}+1}_{\preceq}[w]_{\sim} & := & Min^{\overline{n}}_{\preceq}[w]_{\sim} & \text{if } Min^n_{\preceq}([w]_{\sim}) = [w]_{\sim} \\
Min^{\overline{n}+1}_{\preceq}[w]_{\sim} & := & Min^{\overline{n}}_{\preceq}[w]_{\sim} \cup Min_{\preceq}([w]_{\sim} \setminus Min^n_{\preceq}[w]_{\sim}) & \text{otherwise}
\end{array}
$$

We now can define the logic of knowledge and degrees of belief.

[4] This definition is conditional to modally definable subsets, unlike [Baltag and Smets, 2008, Stalnaker, 1996] where it is on any subset. In that case safe belief is not bisimulation invariant and increases the expressivity of the logic.

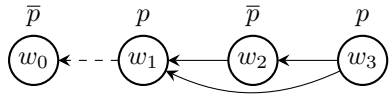


Figure 4.3: A plausibility model wherein the two p worlds are not bisimilar, because they have different higher-order belief properties.

Definition 4.18 (Degrees of belief)**.** We replace the clause for conditional belief in the inductive language definition by a clause $B^n\varphi$ for belief in φ to degree n, for $n \in \mathbb{N}$. The semantics are

$$\mathcal{M}, w \models B^n\varphi \text{ iff for all } v \in Min^n_{\preceq}([w]_\sim) : \mathcal{M}, v \models \varphi$$

In an extended version of this paper we are confident that we will prove that the logics of conditional belief and knowledge, of degrees of belief and knowledge, and both with the addition of safe belief are all expressively equivalent.

4.4 Multi-agent epistemic doxastic logic

For a finite set A of agents and a set of propositional symbols P the *multi-agent epistemic-doxastic language* $\mathcal{L}_{P,A}$ is

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_a\varphi \mid B^\varphi_a\varphi,$$

where $p \in P$ and $a \in A$. *Epistemic plausibility models* are generalized similarly, we now have plausibility relations $\leq_a$ and epistemic relations $\sim_a$ for each agent a. For each agent the domain is partitioned into (possibly) various equivalence classes, such that each class is a well-preorder. The single-agent results do not simply transfer to the multi-agent stage. We give an example.

Example 4.19. Consider Figure 4.3. The solid arrows represent the plausibilities for agent a and the dashed arrow for agent b. In our example, the partition for a is $\{w_0\}, \{w_1, w_2, w_3\}$, whereas the partition for b is $\{w_0, w_1\}, \{w_2\}, \{w_3\}$. Unlike before, the two p-states are not bisimilar, because in the state w_1 agent b is uncertain about the value of p but defeasibly believes p (there is a less plausible alternative w_0, whereas in state w_3 agent b knows (and believes) that p. In both worlds, of course, agent a still believes that p, but a distinguishing formula between the two is now, for example, $\neg K_b p \wedge B_a p$, true in w_1 but false in w_3.

It will be clear from Example 4.19 that we cannot, for each agent, derive a normal plausibility relation $\preceq_a$ from a given plausibility relation $\leq_a$ by identifying

worlds with the same valuation: $w \approx_a w'$ iff $V(w) = V(w')$ and $w \sim_a w'$ does not work (worlds w_1 and w_3 in Example 4.3 satisfy different formulas). Some strengthening guarantees that bisimilarity still implies modal equivalence. An example is, using the above $\approx_a$:

$$\begin{array}{lll} w \approx w' & \text{iff} & (\text{for all agents } a : w \approx_a w') \\ w \preceq_a w' & \text{iff} & (Min_{\leq_a}[w]_{\approx} \leq_a Min_{\leq_a}[w']_{\approx}) \end{array}$$

Unfortunately we do not get that modal equivalence then implies bisimilarity. The strongest possible approach is of course to require that [$w \approx w'$ iff (w, w') is a pair in the bisimulation relation]. This works, but it is is rather self-defeating. In due time we hope to find a proper generalisation in between these two extremes.

4.5 Planning

In planning an agent is tasked with finding a course of action (i.e. a plan) that achieves a given goal. A planning problem implicitly represents a state-transition system, where transitions are induced by actions. Exploring this state-space is a common method for reasoning about and synthesising plans. A growing community investigates planning in dynamic epistemic logic [Bolander and Andersen, 2011, Löwe et al., 2011, Aucher, 2012, Andersen et al., 2012], and using the framework presented here we can in similar fashion consider planning with doxastic attitudes. To this end we identify states with plausibility models, and the goal with a formula of the epistemic doxastic language. Further we can describe the dynamics of actions by using e.g. hard announcements or soft announcements [van Benthem, 2007], or yet more expressive notions such as event models [Baltag and Smets, 2008].

With the state-space consisting of plausibility models, model theoretic results become pivotal to the development of planning algorithms. In general, we cannot require even single-agent plausibility models (even on a finite set of propositional symbols) to be finite. Also, normal plausibility models need not be finite — obvious, as the 'normalising' procedure in which we replace $\leq$ by $\preceq$ does not change the domain. Our definition of bisimulation has a crucial property in this regard: By Lemma 4.11 the bisimulation contraction of a model will contain no two worlds with the same valuation, hence any bisimulation minimal model on a finite set of propositions is finite. Moreover, two bisimulation minimal models are bisimilar exactly when they are isomorphic, and it follows that are only finitely many distinct bisimulation minimal epistemic plausibility models. With the reasonable assumption that actions preserve bisimilarity (this is the case for the types of actions mentioned above), our investigations on the proper

notion of bisimulation therefore allow us to employ a smaller class of models in planning. This is a chief motivation for our work here, and an immediate consequence is that determining whether there exists a plan for a plausibility planning problem is *decidable* (see [Andersen et al., 2014]).

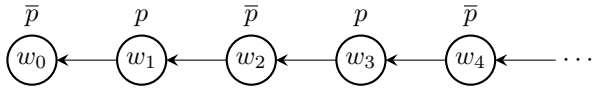


Figure 4.4: Uncontractable chain of p and $\neg p$-worlds.

It is remarkable that the approach of [Demey, 2011] to defining bisimulation for epistemic plausibility models does not yield decidability of planning problems, not even for single-agent models defined on a single proposition. It has, for instance, that the model in Figure 4.4 consisting of an infinite 'directed chain' of alternating p and $\neg p$ worlds (a copy of the natural numbers axis) is bisimulation minimal. In our approach the bisimulation minimal model would be the middle one of Figure 4.1, regardless of the number of worlds. Though [Demey, 2011] also shows that bisimilarity implies modal equivalence and vice versa (for image finite models), this is not inconsistent with our results here. Another difference between our approach and [Demey, 2011] lies in the semantics of safe belief. There, safe belief is relative to any subset (see also Footnote 4). For a 'directed chain' model, the safe belief semantics of [Demey, 2011] permits counting the number of p and $\neg p$ worlds. Such more expressive semantics naturally come at a cost, namely having no finite bound on the size of minimal single-agent models.

Acknowledgements

We thank the reviewers of the AI 13 conference for their comments. Hans van Ditmarsch is also affiliated to IMSc (Institute of Mathematical Sciences), Chennai, as research associate. He acknowledges support from European Research Council grant EPS 313360.

Chapter 5

Addendum to Bisimulation for single-agent plausibility models

Section 5.2 represents joint work between Mikkel Birkegaard Andersen, Thomas Bolander, Hans van Ditmarsch and this author. This work was conducted during a research visit in Nancy, October 2013, and constitutes part of a joint journal publication not yet finalized. The presentation of the results in Section 5.2 is solely the work of this author. The other parts of this chapter are solely due to the author of this thesis.

In the previous chapter we presented a notion of bisimulation for single-agent plausibility models, and showed it to correspond with modal equivalence for the doxastic language containing conditional belief. We further presented semantics for degrees of belief and safe belief, and briefly mentioned the role of model theory in planning — this chapter expands on exactly these topics. In Section 5.1 we show that the notion of bisimulation in Definition 4.9 also corresponds to modal equivalence for the doxastic language containing degrees of belief. Following this, we use Section 5.2 to investigate the relative expressive power of various doxastic languages, separated by which doxastic modalities are included. We conclude on our findings in Section 5.3, and additionally discuss further work.

As we will consider different languages interpreted on plausibility models, we start things off by giving them distinct names. The language $\mathcal{L}_P$ as it was named in Chapter 4 is from this point on referred to as $\mathcal{L}_P^C$, as it adds the conditional doxastic modality to the epistemic language. By $\mathcal{L}_P^{C,\Box}$ we denote the addition of the safe belief modality $\Box$ to $\mathcal{L}_P^C$, and in the usual way we define the dual $\Diamond\varphi$ as $\neg\Box\neg\varphi$. The semantics of safe belief is given below, where we also spell out the semantics of its dual for convenience.

$$\mathcal{M}, w \models \Box\varphi \text{ iff } (\mathcal{M}, w \models B^{\psi}\varphi \text{ for all } \psi \in \mathcal{L}_P^C \text{ such that } \mathcal{M}, w \models \psi)$$
$$\mathcal{M}, w \models \Diamond\varphi \text{ iff } (\mathcal{M}, w \models \widehat{B}^{\psi}\varphi \text{ for some } \psi \in \mathcal{L}_P^C \text{ such that } \mathcal{M}, w \models \psi)$$

Turning to our quantitative doxastic modality, we let $\mathcal{L}_P^D$ denote the epistemic language with the addition of B^n, referred to as either the degree of belief modality or quantitative doxastic modality. For some $k \in \mathbb{N}$ we let $\mathcal{L}_P^{D,k} \subset \mathcal{L}_P^D$ denote the language in which the maximum degree of the quantitative doxastic modality is k.

5.1 Degrees of Belief

In this section we show how the quantitative doxastic modality matches up against the notion of bisimulation given in Definition 4.9. First we establish some results regarding the relationship between $Min_{\preceq}^n$ and the normal plausibility relation. Next we apply these these results to show that the bisimulation found Definition 4.9 corresponds to modal equivalence for $\mathcal{L}_P^D$ (see Definition 3.3) in the typical sense.

5.1.1 Belief Spheres and the Normal Plausibility Relation

For this subsection we let $\mathcal{M} = (W, \leq, \sim, V)$ denote a plausibility model and $\preceq$ the normal plausibility relation of $\mathcal{M}$. As $\sim$ is universal it is immediate that $[w]_{\sim} = W$ for any $w \in W$. Therefore $Min_{\preceq}^n[w]_{\sim}$ is exactly $Min_{\preceq}^n W$, so for notational simplicity we use the latter form exclusively. We refer to the set of worlds $Min_{\preceq}^n W$ as *(belief) sphere* n, and accordingly for $w \in Min_{\preceq}^n W$ we say that w is in (belief) sphere n. The reading of $B^n\varphi$ is that φ is believed to degree n, where degrees should be determined in accordance with the normal plausibility relation $\preceq$. As the semantics are given in terms of $Min_{\preceq}^n$, we must show each belief sphere is in accord with $\preceq$. By this we mean that there is no world w in sphere n, such that any world v *not* in sphere n is at least as

plausible as w; i.e. $v \preceq w$. Having shown this (see Proposition 5.2), we have another, equivalent, reading of $B^n\varphi$, namely that φ is true in every world of sphere n.

Lemma 5.1. *Belief sphere n is a subset of belief sphere $n+1$.*

Proof. Immediate from the recursive definition in Section 4.3. □

Proposition 5.2. *Any world in sphere n is strictly more plausible wrt. $\preceq$ than any world not in sphere n.*

Proof. Given worlds $w, v \in W$ and any $n \in \mathbb{N}$. We show that if $w \in Min^n_{\preceq} W$ and $v \notin Min^n_{\preceq} W$, then $w \prec v$. We proceed by induction on n. For the base case assume that $w \in Min^0_{\preceq} W$ and $v \notin Min^0_{\preceq} W$. By $\preceq$-minimality we have that $w \preceq u$ for all $u \in W$, and so also $w \preceq v$. Assuming towards a contradiction that $v \preceq w$, we have from $\preceq$-transitivity that $v \preceq u$ for all $u \in W$. But this leads to the contradiction that $v \in Min^0_{\preceq} W$, and so we conclude $v \npreceq w$. We now have that $w \prec v$, settling the base case.

Assume now that $w \in Min^{n+1} W$ and $v \notin Min^{n+1}_{\preceq} W$. By assumption sphere $n+1$ must be a proper subset of W, so we have that $w \in (Min^n_{\preceq} W \cup Min_{\preceq}(W \setminus Min^n_{\preceq} W))$, which gives rise to two cases. If $w \in Min^n_{\preceq} W$ then it follows from Lemma 5.1 that $v \notin Min^n_{\preceq} W$, and so we have $w \prec v$ by applying the induction hypothesis. Otherwise we have $w \in Min_{\preceq}(W \setminus Min^n_{\preceq} W)$, and so from $\preceq$-minimality it follows that $w \preceq u$ for all $u \in (W \setminus Min^n_{\preceq} W)$. With our initial assumption we have $v \in (W \setminus (Min^n_{\preceq} W \cup Min_{\preceq}(W \setminus Min^n_{\preceq} W)))$, which implies that $v \in (W \setminus Min^n_{\preceq} W)$ and consequently $w \preceq v$. As in the base case the assumption of $v \preceq w$ leads to a contradiction, because we have $v \notin Min_{\preceq}(W \setminus Min^n_{\preceq} W)$. As $v \npreceq w$ we can conclude that $w \prec v$. □

Recall that the definition of minimality gives us that if $w \in Min_{\preceq} W$ then $w \preceq v$ for any $v \in W$. However, this only holds for sphere 0; consider for instance a plausibility model with two worlds w, v where $w \prec v$, we have $v \in Min^1_{\preceq} W$ but at the same time $v \npreceq w$. The issue here is that sphere 1 is a proper superset of sphere 0. For worlds that just made the cut into a sphere, we have the following useful analogue.

Lemma 5.3. *Given any $n \in \mathbb{N}$ and any $w \in W$. If $w \notin Min^n_{\preceq} W$, then $v \preceq w$ for any $v \in Min^{n+1}_{\preceq} W$.*

Proof. We consider two exhaustive cases for v. Either $v \in (Min^{n+1}_{\preceq} W \setminus Min^n_{\preceq} W)$ in which case $v \preceq w$ follows from minimality, since by assumption $w \in (W \setminus Min^n_{\preceq} W)$. Otherwise $v \in Min^n_{\preceq} W$, and so from $w \notin Min^n_{\preceq} W$ and Proposition 5.2 it follows that $v \prec w$ and hence also $v \preceq w$. □

Lemma 5.4. *Worlds that are $\preceq$-equiplausible belong to exactly the same belief spheres.*

Proof. We show that if $v \preceq w$ and $w \preceq v$, then for any $n \in \mathbb{N}$, $w \in Min^n_{\preceq} W$ iff $v \in Min^n_{\preceq} W$. As $\preceq$ is a total preorder then by the assumption of $v \preceq w$ and $w \preceq v$, it follows by minimality that for any subset $W' \subseteq W : w \in Min_{\preceq} W'$ iff $v \in Min_{\preceq} W'$. Expanding the definitions of $Min^n_{\preceq} W$ and $Min^n_{\preceq} W$ we see that both are unions of such $\preceq$-minimal subsets of W, hence it follows that $w \in Min^n_{\preceq} W$ iff $v \in Min^n_{\preceq} W$. □

Lemma 5.5. *Worlds that have the same valuation belong to exactly the same belief spheres.*

Proof. Consider $w, v \in W$ where $V(w) = V(v)$. Then $w \approx v$ and thus $Min_{\leq}[w]_{\approx} = Min_{\leq}[v]_{\approx}$. By Definition 4.6 we have $w \preceq v$ and $v \preceq w$, hence applying Lemma 5.4 we arrive at the conclusion. □

5.1.2 Bisimulation and Modal Equivalence for $\mathcal{L}^D_P$

We now show that the notion of bisimulation is proper for plausibility models when considering the language containing the quantitative doxastic modality.

Lemma 5.6. *Let two plausibility models $\mathcal{M} = (W, \leq, \sim, V)$, $\mathcal{M}' = (W', \leq', \sim', V')$ with $w \in W$ and $w' \in W'$ be given, and let $\preceq$, $\preceq'$ denote their respective normal plausibility relations. If $\mathfrak{R}$ is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ where $(w, w') \in \mathfrak{R}$, then for any $n \in \mathbb{N}$, $w \in Min^n_{\preceq} W$ iff $w' \in Min^n_{\preceq'} W'$.*

Proof. Due to symmetry it is sufficient to show that $w \in Min^n_{\preceq} W$ implies $w' \in Min^n_{\preceq'} W'$. We proceed by induction on n.

To show the base case let $w \in Min^0_{\preceq} W$. Towards a contradiction assume that $w' \notin Min^0_{\preceq} W'$. Applying Proposition 5.2 this means there is a $v' \in Min^0_{\preceq'} W'$ s.t. $v' \preceq' w'$ and $w' \not\preceq' v'$. From [back$_{\preceq}$] we have a $v \in W$ s.t. $v \preceq w$ and $(v, v') \in \mathfrak{R}$. As w is $\preceq$-minimal in W we must have $w \preceq v$. By [forth$_{\preceq}$] there is $u' \in W'$ s.t. $u' \preceq' v'$ and $(w, u') \in \mathfrak{R}$, and since v' is $\preceq'$-minimal we have $u' \in Min^0_{\preceq} W'$. By [atoms] it follows that $V(w) = V'(w') = V'(u')$, and so as $w' \approx' u'$ we have by Lemma 5.5 that $w' \in Min^0_{\preceq} W'$, a contradiction.

For the induction step assume that $w \in Min^{k+1}_{\preceq} W$ and let $V = (W \setminus Min^k_{\preceq} W)$. With this definition of V we have that $Min^{k+1}_{\preceq} W = (Min^k_{\preceq} W) \cup (Min_{\preceq} V)$ and further that $(Min^k_{\preceq} W) \cap V$ is the empty set. Therefore either $w \in Min^i_{\preceq} W$

for $0 \leq i \leq k$, or $w \in Min_{\preceq} V$. In the former case we have $w' \in Min^{i}_{\preceq'} W'$ immediately from the induction hypothesis.

We can therefore assume for the remainder that $w \in Min_{\preceq} V$. Now let $V' = (W' \setminus Min^{k}_{\preceq'} W')$ and observe that $Min^{k+1}_{\preceq'} W' = (Min^{k}_{\preceq'} W') \cup (Min_{\preceq'} V')$. Towards a contradiction assume that $w' \notin Min^{k+1}_{\preceq'} W'$. By Proposition 5.2 we therefore have a $v' \in Min_{\preceq'} V'$ s.t. $v' \preceq w'$ and $w' \not\preceq v'$. This also implies that $v' \notin Min^{k}_{\preceq'} W'$. By [back$_{\preceq}$] there is $v \in W$ s.t. $v \preceq w$ and $(v, v') \in \mathfrak{R}$. It cannot be the case that $v \in Min^{k}_{\preceq} W$, because the induction hypothesis would yield the contradiction $v' \in Min^{k}_{\preceq'} W'$, and so we must have $v \in V$. As w is $\preceq$-minimal in V it follows that $w \preceq v$, and so by [forth$_{\preceq}$] there is $u' \in W'$ s.t. $u' \preceq v'$ and $(w, u') \in \mathfrak{R}$. By [atoms] we have that $V(w) = V'(w') = V'(u')$ and therefore $w' \approx' u'$. By assumption of $w' \notin Min^{k+1}_{\preceq'} W'$ we can therefore apply Lemma 5.5 to conclude that $u' \notin Min^{k+1}_{\preceq'} W'$. As v' is in sphere $k+1$, we can apply Proposition 5.2 to conclude $v' \prec u'$. But this contradicts $u' \preceq' v'$, and so it must be the case that $w' \in Min^{k+1}_{\preceq'} W'$. □

Having shown that bisimilar worlds (even in different models) must have the same degree of belief, the following is readily established.

Proposition 5.7. *Bisimilarity implies modal equivalence for* $\mathcal{L}^{D}_{P}$.

Proof. We show that for all $\varphi \in \mathcal{L}^{D}_{P}$, if $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ are bisimilar then $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}', w' \models \varphi$. For $\mathcal{M} = (W, \leq, \sim, V)$ and $\mathcal{M}' = (W, \leq', \sim', V')$, let $\mathfrak{R}$ be a bisimulation s.t. $(w, w') \in \mathfrak{R}$. We proceed by induction on the structure of φ, so by Proposition 4.12 we need only cover the case of $B^{n}\varphi$.

We must show that $\mathcal{M}, w \models B^{n}\varphi$ iff $\mathcal{M}', w' \models B^{n}\varphi$ for any $n \in \mathbb{N}$. By symmetry it suffices to show that $\mathcal{M}, w \models B^{n}\varphi$ implies $\mathcal{M}', w' \models B^{n}\varphi$, which means that $\mathcal{M}', v' \models \varphi$ for all $v' \in Min^{n}_{\preceq'} W'$. So assume that $\mathcal{M}, w \models B^{n}\varphi$. For any $v' \in Min^{n}_{\preceq'} W'$ there will be (by totality of bisimulations) a $v \in W$ s.t. $(v, v') \in \mathfrak{R}$. By Lemma 5.6 it follows that $v \in Min^{n}_{\preceq} W$, and so by assumption of $\mathcal{M}, w \models B^{n}\varphi$ we have $\mathcal{M}, v \models \varphi$. By the induction hypothesis we can conclude $\mathcal{M}', v' \models \varphi$, and so we have $\mathcal{M}', w' \models B^{n}\varphi$. □

As a brief intermission we remark that both $\sim$ and $\leq$ are primitives in our plausibility models. So if $\mathcal{M}$ is image-finite we require that for any world $w \in \mathcal{M}$, the sets $\{v \in W \mid v \sim w\}$ and $\{v \in W \mid v \leq w\}$ must be finite [Blackburn et al., 2001]. As $\sim$ is universal this has the very significant implication (which is generally not the case for Kripke models) that any image-finite plausibility models is in fact *finite*! Consequently we have that any belief sphere of an image-finite model is finite. We return to the business at hand, showing the

prototypical other direction of Proposition 5.7. To this end we first provide the counterpart to Lemma 5.6 in the case of image-finite and modally equivalent models for $\mathcal{L}_P^D$ (for the notion of modal equivalence, see Definition 3.3).

Lemma 5.8. *Let image-finite pointed plausibility models $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ where $\mathcal{M} = (W, \leq, \sim, V)$ and $\mathcal{M}' = (W', \leq', \sim', V')$ be given. Further let $\preceq$, $\preceq'$ denote their respective normal plausibility relations. If $(\mathcal{M}, w) \equiv_{\mathcal{L}_P^D} (\mathcal{M}', w')$, then for any $n \in \mathbb{N}$ it is the case that $w \in Min^n_{\preceq} W$ iff $w' \in Min^n_{\preceq'} W'$.*

Proof. Due to symmetry we need only show left to right, so assume that $w \in Min^n_{\preceq} W$. Towards a contradiction assume that $w' \notin Min^n_{\preceq'} W'$. For any $v' \in Min^n_{\preceq'} W'$ we therefore have that v' is not in the same belief spheres as w', so by the contrapositive of Lemma 5.5 it follows that $V'(w') \neq V'(v')$. Furthermore, as $(\mathcal{M}, w) \equiv_{\mathcal{L}_P^D} (\mathcal{M}', w')$ it follows that $V(w) = V'(w')$, and therefore also $V(w) \neq V'(v')$.

Having established this fact means we can find a simple formula $\gamma_{w,v'}$ for w and v', which is true in w but false in v'. Formally, if there is a $p \in (V(w) \setminus V'(v'))$ then $\gamma_{w,v'} = p$, and otherwise $\gamma_{w,v'} = \neg q$ for some $q \in (V'(v') \setminus V(w))$. We define

$$\Gamma^n_w = \bigwedge_{v' \in Min^n_{\preceq'} W'} \gamma_{w,v'}$$

which is finite by image finiteness of $\mathcal{M}'$. By construction we have that Γ^n_w holds in w, but not in any world belonging to belief sphere n of $\mathcal{M}'$. In symbols this means that $\mathcal{M}, w \models \Gamma^n_w$ and by assumption of $w \in Min^n_{\preceq} W$ it follows that $\mathcal{M}, w \models \widehat{B}^n \Gamma^n_w$. Furthermore, because there is no $v' \in Min^n_{\preceq'} W'$ s.t. $\mathcal{M}', v' \models \Gamma^n_w$, we can conclude that $\mathcal{M}', w' \not\models \widehat{B}^n \Gamma^n_w$. But this contradicts $(\mathcal{M}, w) \equiv_{\mathcal{L}_P^D} (\mathcal{M}', w')$ and consequently it must be the case that $w' \in Min^n_{\preceq'} W'$ □

Having established this result, the following is shown using the same proof technique as in the Hennessy-Milner theorem.

Proposition 5.9. *On the class of image-finite models, modal equivalence for $\mathcal{L}_P^D$ implies bisimilarity.*

Proof. Let $\mathcal{M} = (W, \leq, \sim, V)$ and $\mathcal{M}' = (W', \leq', \sim', V')$ be two image-finite, plausibility models on P, and define $\mathfrak{R} \subseteq W \times W'$, such that $(w, w') \in \mathfrak{R}$ iff $(\mathcal{M}, w) \equiv_{\mathcal{L}_P^D} (\mathcal{M}', w')$. We show that $\mathfrak{R}$ satisfies the clauses in Definition 4.9, meaning that the $\equiv_{\mathcal{L}_P^D}$ relation itself is a bisimulation. Foregoing [atoms], [forth$_\sim$] and [back$_\sim$], we will prove the most involved case [forth$_\preceq$] ([back$_\preceq$] can be shown symmetrically).

Assume that $(\mathcal{M}, w) \equiv_{\mathcal{L}_P^D} (\mathcal{M}', w')$, $v \in W$ and $v \preceq w$. Towards a contradiction assume that there is no $v' \in W'$ such that $v' \preceq' w'$ and $(\mathcal{M}, v) \equiv_{\mathcal{L}_P^D} (\mathcal{M}', v')$. Let $S' = \{v' \in W' \mid v' \preceq' w'\} = \{v'_1, \dots v'_m\}$ be the $\preceq'$-successors of w' (S' is finite by image-finiteness of $\mathcal{M}'$). By assumption every $v'_i \in S'$ is modally inequivalent to v, so for each v'_i we have a formula φ_i s.t. $\mathcal{M}, v \models \varphi_i$ and $\mathcal{M}', v'_i \not\models \varphi_i$. Letting $\Phi = \varphi_1 \wedge \dots \wedge \varphi_m$ it follows that $\mathcal{M}, v \models \Phi$ and $\mathcal{M}', v'_i \not\models \Phi$.

If $w \in Min^0_{\preceq} W$ then $v \preceq w$ means that $v \in Min^0_{\preceq} W$, and so $\mathcal{M}, w \models \widehat{B}^0\Phi$. Further, it follows from Lemma 5.8 that $w' \in Min^0_{\preceq'} W'$. By $\preceq$-minimality of w' in W' we have therefore that $S' = Min^0_{\preceq'} W'$. By construction of Φ this means that $\mathcal{M}', w' \not\models \widehat{B}^0\Phi$, contradicting the assumption that w and w' are modally equivalent.

If $w \notin Min^0_{\preceq} W$, let n be chosen such that $w \in Min^{n+1}_{\preceq} W$ and $w \notin Min^n_{\preceq} W$. Such an n exists as W is finite. Since $v \preceq w$, it must be the case that $v \in Min^{n+1}_{\preceq} W$ as Lemma 5.2 would otherwise imply $w \prec v$. Therefore from $\mathcal{M}, v \models \Phi$ we have that $\mathcal{M}, w \models \widehat{B}^{n+1}\Phi$. Moreover, by Lemma 5.8 it follows that $w' \in Min^{n+1}_{\preceq'} W'$ and $w' \notin Min^n_{\preceq'} W'$. By Lemma 5.3 we can therefore conclude $v' \preceq' w'$ for all $v' \in Min^{n+1}_{\preceq'} W'$ and hence $Min^{n+1}_{\preceq'} W' \subseteq S'$.[1] By construction of Φ this means there are no worlds in $Min^{n+1}_{\preceq'} W'$ satisfying Φ, and so $\mathcal{M}', w' \not\models \widehat{B}^n\Phi$. But this contradicts the assumption of $(\mathcal{M}, w) \equiv_{\mathcal{L}_P^D} (\mathcal{M}', w')$.

Consequently we have that there must exist some $v' \in W'$ with $v' \preceq' w'$ s.t. $(\mathcal{M}, v) \equiv_{\mathcal{L}_P^D} (\mathcal{M}', v')$, showing that $\equiv_{\mathcal{L}_P^D}$ satisfies [forth$_{\preceq}$]. □

By going through the structural characterisation of bisimulation, we now have an immediate correspondence between modal equivalence for the language with conditional beliefs and modal equivalence for the language with degrees of belief.

Corollary 5.10. *On the class of image-finite models, modal equivalence for $\mathcal{L}_P^C$ corresponds to modal equivalence for $\mathcal{L}_P^D$.*

Proof. Left to right follows from Proposition 4.16 and Proposition 5.7. Right to left follows similarly from 5.9 and Proposition 4.12. □

5.2 Expressive power

This section represents joint work between Mikkel Birkegaard Andersen,

[1] While sphere n is in fact exactly S', this is not necessary to show for this proof.

Thomas Bolander, Hans van Ditmarsch and this author. This work was conducted during a research visit in Nancy, October 2013, and constitutes part of a joint journal publication not yet finalized. The presentation of the results in this section is solely the work of this author.

Since we're investigating different logical languages interpreted on the same class of models, there's more to the story than bisimulation and modal equivalence. As we saw already in Example 4.13, having conditional belief allow us to distinguish models which cannot be distinguished using only unconditional belief. In this sense conditional belief can express more properties of a model than unconditional belief. Adopting this view of formulas we can for instance say that $B^p q$ describes a certain property of plausibility models, namely that of plausibility models in which, when given information that p is true, q is believed. Now we can think of all the properties expressible using formulas of $\mathcal{L}_P^C$ as a benchmark of how expressive the language with conditional belief really is. While gauging expressivity in absolute terms might provide further insight into a language, our main focus will be on the *relative* expressive power between $\mathcal{L}_P^C$ and the languages introduced above with different doxastic modalities.

In the next section we formally introduce the notion of expressive power, as well as the notion of parameterized bisimulation. In Section 5.2.2 we show that the language with conditional belief and the languages with degrees of belief are incomparable, and in Section 5.2.3 we show that adding safe belief to the language with conditional belief yields a strictly more expressive language. This is in fact opposite of what we we're confident could be proven at the end of Section 4.3 (a carbon-copy of our original publication [Andersen et al., 2013]).

5.2.1 Formalities

We adopt the notions and basic concepts of the expressive power (or, expressivity) of a language as presented in [van Ditmarsch et al., 2007]. We remind the reader that $\varphi_1 \equiv \varphi_2$ means that φ_1 and φ_2 are equivalent as per Definition 3.2. While $\equiv$ is overloaded below, this introduces no ambiguity, as it will be clear from context whether we're referring to models, formulas or languages.

Definition 5.11. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages interpreted in the same class of models.

- $\mathcal{L}_2$ is *at least as expressive* as $\mathcal{L}_1$ (denoted $\mathcal{L}_1 \leqq \mathcal{L}_2$) iff for every $\varphi_1 \in \mathcal{L}_1$ there is a $\varphi_2 \in \mathcal{L}_2$ s.t. $\varphi_1 \equiv \varphi_2$.

- $\mathcal{L}_1$ and $\mathcal{L}_2$ are *equally expressive* (denoted $\mathcal{L}_1 \equiv \mathcal{L}_2$) iff $\mathcal{L}_1 \leqq \mathcal{L}_2$ and $\mathcal{L}_2 \leqq \mathcal{L}_1$.
- $\mathcal{L}_2$ is *more expressive* than $\mathcal{L}_1$ (denoted $\mathcal{L}_1 < \mathcal{L}_2$) iff $\mathcal{L}_1 \leqq \mathcal{L}_2$ and $\mathcal{L}_2 \not\leqq \mathcal{L}_1$.
- $\mathcal{L}_1$ and $\mathcal{L}_2$ are *incomparable* iff $\mathcal{L}_1 \not\leqq \mathcal{L}_2$ and $\mathcal{L}_2 \not\leqq \mathcal{L}_1$.

We will be showing several cases below where $\mathcal{L}_1 \not\leqq \mathcal{L}_2$. Our modus operandi is to show there is a $\varphi_1 \in \mathcal{L}_1$, where for any $\varphi_2 \in \mathcal{L}_2$ we can find two pointed models $(\mathcal{M}, w), (\mathcal{M}', w')$ such that $\mathcal{M}, w \models \varphi_1$, $\mathcal{M}', w' \not\models \varphi_1$ and $(\mathcal{M}, w \models \varphi_2 \Leftrightarrow \mathcal{M}', w' \models \varphi_2)$. In other words, for some $\varphi_1 \in \mathcal{L}_1$, no matter the choice of $\varphi_2 \in \mathcal{L}_2$, there will be models which φ_1 distinguishes but φ_2 does not, meaning that $\varphi_1 \not\equiv \varphi_2$.

It becomes helpful to show that plausibility models are modally equivalent, when we do not allow certain propositional symbols to occur in formulas. To this end we introduce restricted bisimulations, meaning that only a subset of atoms are considered. Technically, for some plausibility model $\mathcal{M} = (W, \leq, \sim, V)$ on P we consider any $Q \subseteq P$. Here Q is the set of atoms we "care about", and so a restricted equivalence relation on W is given by

$$w \approx^Q w' \quad \text{iff} \quad (V(w) \cap Q) = (V(w') \cap Q)$$

Immediately we can define a restricted version of the normal plausibility relation, viz.

$$w \preceq^Q w' \quad \text{iff} \quad Min_{\leq}[w]_{\approx^Q} \leq Min_{\leq}[w']_{\approx^Q}$$

For $\approx^P$ and $\preceq^P$ we will omit P as index, as these relation are exactly the non-restricted definitions of $\approx$ and $\preceq$ presented in Section 4.2.2.

Definition 5.12 (Restricted Bisimulation)**.** Let plausibility models $\mathcal{M} = (W, \leq, \sim, V)$ and $\mathcal{M}' = (W', \leq', \sim', V')$ on P be given, and further let $Q \subseteq P$. Denote by $\preceq^Q, \preceq'^Q$ the respective derived normal plausibility relations. A non-empty relation $\mathfrak{R} \subseteq W \times W'$ is a *Q-bisimulation* between $\mathcal{M}$ and $\mathcal{M}'$ if for all $(w, w') \in \mathfrak{R}$:

[atoms] $(V(w) \cap Q) = (V'(w') \cap Q)$.

[forth$_\preceq$] If $v \in W$ and $v \preceq^Q w$, there is a $v' \in W'$ s.t. $v' \preceq'^Q w'$ and $(v, v') \in \mathfrak{R}$.

[back$_\preceq$] If $v' \in W'$ and $v' \preceq'^Q w'$, there is a $v \in W$ s.t. $v \preceq^Q w$ and $(v, v') \in \mathfrak{R}$.

[forth$_\sim$], [back$_\sim$] as in Definition 4.9

If a Q-bisimulation between $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ exists, the two models are called *Q-bisimilar*.

This definition yields the following result.

Proposition 5.13. *Q-bisimilarity implies modal equivalence for $\mathcal{L}_Q^C$.*

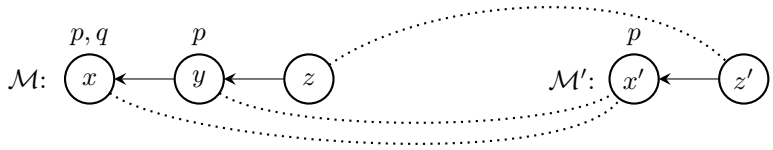
Proof. Almost exactly as in Proposition 4.12, the difference being when $\varphi \in \mathcal{L}_Q^C$ is propositional, meaning that $\varphi = q$ for some $q \in Q$. The truth value of q must coincide for any pair of worlds in the bisimulation relation due to [atoms] (in the restricted bisimulation). □

5.2.2 Expressive Power of Degrees of Belief

We now treat the relative expressive power of $\mathcal{L}_P^C$ and $\mathcal{L}_P^D$. Assuming that P is countably *infinite* we can show that these two languages are in fact incomparable.

Proposition 5.14. *If P is countably infinite, then $\mathcal{L}_P^D \not\leqslant \mathcal{L}_P^C$.*

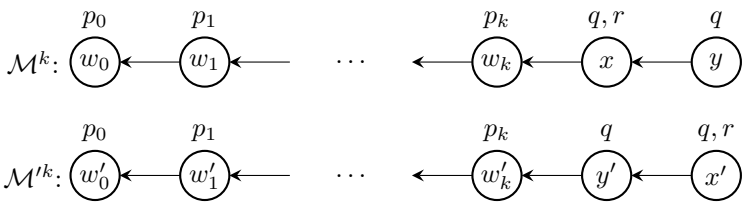
Proof. Let $p \in P$ and consider the formula $B^1 p$ which belongs to $\mathcal{L}_P^D$, and further take any formula $\varphi_C \in \mathcal{L}_P^C$. As φ_C is finite, there will be *some* q in the countably infinite set P which does not occur in φ_C. It follows then that $\varphi_C \in \mathcal{L}_{P\setminus\{q\}}^C$. For any such q there exists two models of the form below, where we have that $\mathfrak{R} = \{(x, x'), (y, x'), (z, z')\}$ is a $(P \setminus \{q\})$-bisimulation; the dotted edges illustrate $\mathfrak{R}$.



Since $(x, x') \in \mathfrak{R}$ it follows from Proposition 5.13 that $(\mathcal{M}, x) \equiv_{\mathcal{L}_{P\setminus\{q\}}^C} (\mathcal{M}', x')$. It is therefore clear that $(\mathcal{M}, x \models \varphi_C)$ iff $(\mathcal{M}', x' \models \varphi_C)$. What is more we have that $\mathcal{M}, x \models B^1 p$ whereas $\mathcal{M}', x' \not\models B^1 p$. This means that using the formula $B^1 p$ of $\mathcal{L}_P^D$, for any formula φ_C of $\mathcal{L}_P^C$ there are models which $B^1 p$ distinguishes but φ_C does not; i.e. $B^1 p \not\equiv \varphi_C$. Consequently we have that $\mathcal{L}_P^D \not\leqslant \mathcal{L}_P^C$. □

Proposition 5.15. *If P is countably infinite, then $\mathcal{L}_P^C \not\leqslant \mathcal{L}_P^D$.*

Proof. We first prove that the two models below, defined for any $k \in \mathbb{N}$, are modally equivalent for a subset of $\mathcal{L}_P^D$. We stress the importance of the fact that x (resp. y) has the same valuation as x' (resp. y'), and that x is strictly more plausible than y whereas y' is strictly more plausible than x'.



Claim. For any $k \in \mathbb{N}$, let $\mathcal{M}^k$ and $\mathcal{M}'^k$ be models of the above form, then $(\mathcal{M}^k, w_0) \equiv_{\mathcal{L}_P^{D,k}} (\mathcal{M}'^k, w'_0)$.

Proof of claim. Recall that $\mathcal{L}_P^{D,k}$ is the language in which the degree of any quantitative belief modality is at most k. We will prove the stronger version of this claim, namely that $(\mathcal{M}^k, w_i) \equiv_{\mathcal{L}_P^{D,k}} (\mathcal{M}'^k, w'_i)$ for $0 \leq i \leq k$, $(\mathcal{M}^k, x) \equiv_{\mathcal{L}_P^{D,k}} (\mathcal{M}'^k, x')$ and $(\mathcal{M}^k, y) \equiv_{\mathcal{L}_P^{D,k}} (\mathcal{M}'^k, y')$.

We let $\varphi \in \mathcal{L}_P^{D,k}$ and proceed by induction on φ. For the base case φ is a propositional symbol, and so as the valuation of each w_i matches that of w'_i, x matches x' and y matches y' we're done. The cases of negation and conjunction readily follow using the induction hypothesis.

Consider now $\varphi = K\psi$ and assume that $\mathcal{M}^k, w_0 \models K\psi$. This means that for all $v \in \{w_0, \ldots, w_k, x, y\} : \mathcal{M}^k, v \models \psi$. By the induction hypothesis we therefore have for all $v' \in \{w'_0, \ldots, w'_k, y', x'\} : \mathcal{M}'^k, v' \models \psi$, and hence $\mathcal{M}'^k, w'_0 \models K\psi$. The exact same argument holds for $w_1, \ldots, w_k, x, y$, and the other direction (right to left) is symmetrical.

Consider $\varphi = B^j\psi$ for $0 \leq j \leq k$, and recall that this is sufficient since $\varphi \in \mathcal{L}_P^{D,k}$. Let W and W' respectively denote the domain of $\mathcal{M}^k$ and $\mathcal{M}^k$. As every world in the above models have different valuations, both $\mathcal{M}^k$ and $\mathcal{M}'^k$ are normal, that is, the plausibility relation and the normal plausibility relation are identical. Therefore $Min^j_{\preceq} W = \{w_0, \ldots, w_j\}$ and $Min^j_{\preceq'} W' = \{w'_0, \ldots, w'_j\}$. Assuming $\mathcal{M}^k, w_0 \models B^j\psi$ we have that $\mathcal{M}^k, v \models \psi$ for all $v \in Min^j_{\preceq} W$, and so this means that $\mathcal{M}, v \models \psi$ for all $v \in \{w_0, \ldots, w_j\}$. As $j \leq k$ we can apply the induction hypothesis to conclude $\mathcal{M}'^k, v' \models \psi$ for all $v' \in \{w'_0, \ldots, w'_j\} = Min^j_{\preceq'} W'$. From

this it follows that $\mathcal{M}'^k, w_0' \models B^j\psi$ as required. The same argument applies to $w_1, \ldots, w_k$, x, y because $Min^j_{\preceq} W$ (i.e. belief sphere j) is unaltered by the world in which $B^j\psi$ is evaluated (importantly, x and y are *not* in belief sphere k). The other direction is shown symmetrically thus completing the proof of this claim.

Consider now $B^q r$ belonging to $\mathcal{L}_P^C$ and any formula $\varphi_D \in \mathcal{L}_P^D$. Since φ_D is finite there is some $k \in \mathbb{N}$ such that $\varphi_D \in \mathcal{L}_P^{D,k}$ (e.g. take k to be the largest degree of belief modality occurring in φ_D). By the above claim we therefore have models $\mathcal{M}^k$ and $\mathcal{M}'^k$ such that $(\mathcal{M}^k, w_0 \models \varphi_D) \Leftrightarrow (\mathcal{M}'^k, w_0' \models \varphi_D)$. The reason such models always exist is that $p_0, \ldots, p_k$ are taken from a countably infinite set of symbols and $k \in \mathbb{N}$.

To determine the truth of $B^q r$ we point out that $[\![q]\!]_{\mathcal{M}^k} = \{x, y\}$ and $[\![q]\!]_{\mathcal{M}'^k} = \{y', x'\}$. Given the plausibility relation we have that conditional on q, x is minimal in $\mathcal{M}^k$ and y' is minimal in $\mathcal{M}'^k$. Since $\mathcal{M}^k, x \models r$ and $\mathcal{M}'^k, y' \not\models r$, it follows that $\mathcal{M}^k, w_0 \models B^q r$, while $\mathcal{M}'^k, w_0' \not\models B^q r$. With this we have shown that taking the formula $B^q r$ of $\mathcal{L}_P^C$, there is for any $\varphi_D \in \mathcal{L}_P^D$ models which $B^q r$ distinguishes but φ_D does not; i.e. $B^q r \not\equiv \varphi_D$. It follows that $\mathcal{L}_P^C \not\leqslant \mathcal{L}_P^D$. □

The two propositions above yield the following result.

Corollary 5.16. *If P is countably infinite, the languages $\mathcal{L}_P^C$ and $\mathcal{L}_P^D$ are incomparable.*

The proof of Proposition 5.14 illustrates the inability of conditional belief to capture quantitative belief (even to degree 1), because the latter represents belief that is irrespective of some chosen facts. The proof of Proposition 5.15 showcases the ability of conditional belief to capture a form of quantitative belief to an *arbitrary degree*, with this degree depending on factual content (here simply p) rather than a fixed number. In both proofs it is important that an infinite number of symbols are at our disposal. How the two languages compare when P is finite has not been settled.

The incomparability of $\mathcal{L}_P^C$ and $\mathcal{L}_P^D$ is remarkable in light of Corollary 5.10, since the incomparability result is established without relying on image-*infinite* models. Rather, we gave simple formulas in both languages that describe an infinite number of models, but which the other (respective) language cannot finitely capture. Taking on a broader perspective this suggests that conditional belief and degrees of belief should not be considered *alternatives*, but rather as complementary doxastic attitudes able to express different properties of plausibility models.

5.2.3 Expressive Power of Safe Belief

As $\mathcal{L}_P^{C,\Box}$ is a superset of $\mathcal{L}_P^C$ it is clear that the former is at least as expressive as the latter, that is, $\mathcal{L}_P^C \leqq \mathcal{L}_P^{C,\Box}$. However, if P is countably infinite, then we can show the two languages to not be equally expressive. The proof is very similar in fashion to the proof of Proposition 5.14.

Proposition 5.17. *If P is countably infinite, then $\mathcal{L}_P^{C,\Box} \nleqq \mathcal{L}_P^C$*

Proof. Let $p \in P$ and consider the formula $\Diamond p$ which belongs to $\mathcal{L}_P^{C,\Box}$, and further let $\varphi_C \in \mathcal{L}_P^C$ be arbitrarily chosen. As φ_C is finite, there will be *some* q in the countably infinite set P not occurring in φ_C, and so $\varphi_C \in \mathcal{L}_{P\setminus\{q\}}^C$. For any such q there exists two models of the form below where we have that $\mathfrak{R} = \{(x,x'),(y,y'),(z,z')\}$ is a $(P \setminus \{q\})$-bisimulation; $\mathfrak{R}$ is indicated with dotted edges.

Since $(z,z') \in \mathfrak{R}$ it follows from Proposition 5.13 that $(\mathcal{M}^q, z) \equiv_{\mathcal{L}_{P\setminus\{q\}}^C} (\mathcal{M}'^q, z')$. We now have that $(\mathcal{M}^q, z \models \varphi_C)$ iff $(\mathcal{M}'^q, z' \models \varphi_C)$ and want to show that, in contrast to φ_C, $\Diamond p$ distinguishes the two pointed models. The semantics of safe belief state that $\mathcal{M}^q, z \models \Diamond p$ iff there is some $\psi \in \mathcal{L}_P^C$ s.t. $(\mathcal{M}^q, z \models \psi$ and $\mathcal{M}^q, z \models \widehat{B}^\psi p)$. From $\mathcal{M}^q, z \models q$ and $\mathcal{M}^q, z \models \widehat{B}^q p$, and since q is a formula of $\mathcal{L}_P^C$, we can conclude $\mathcal{M}^q, z \models \Diamond p$.

We will now show that $\mathcal{M}'^q, z' \not\models \Diamond q$, i.e. that there is *no* $\psi \in \mathcal{L}_P^C$ s.t. $(\mathcal{M}'^q, z' \models \psi$ and $\mathcal{M}'^q, z' \models \widehat{B}^\psi p)$. Assume towards a contradiction that such a ψ did exist. Since $x' \approx' z'$ it follows from Lemma 4.11 and Proposition 4.12 that $(\mathcal{M}'^q, x') \equiv_{\mathcal{L}_P^C} (\mathcal{M}'^q, z')$. From our assumption of ψ we have that $\mathcal{M}'^q, z' \models \psi$, so then it must be the case that $\{x', z'\} \subseteq [\![\psi]\!]$. Inspecting the plausibility relation this means that $Min_{\leq'}[\![\psi]\!] = \{x'\}$. Since $\mathcal{M}'^q, x' \not\models p$ we must therefore also have $\mathcal{M}'^q, z' \not\models \widehat{B}^\psi p$, but this contradicts our initial assumption. Therefore such a ψ cannot exist and we can conclude that $\mathcal{M}'^q, z' \not\models \Diamond p$.

We now have that using the formula $\Diamond p$ of $\mathcal{L}_P^{C,\Box}$, for any formula of $\varphi_C \in \mathcal{L}_P^C$ there are models which $\Diamond p$ distinguishes but φ_C does not; i.e. $\Diamond p \not\equiv \varphi_C$. It

therefore follows that $\mathcal{L}_P^{C,\Box} \not\leq \mathcal{L}_P^C$. □

Corollary 5.18. *If P is countably infinite, then $\mathcal{L}_P^{C,\Box}$ is more expressive than $\mathcal{L}_P^C$.*

Safe belief enables us to express properties about infinitely many propositions with a finite formula, exactly as was the case for degrees of a belief. This is what prompts the increase in expressive power, again under the assumption that we have infinitely many propositional symbols available, even though $\Box$ is given its semantics in terms of conditional belief.

Having established these results, we close with a few remarks about the languages we have not treated. We use the obvious naming scheme for doing so. The languages $\mathcal{L}_P^{\Box}$ and $\mathcal{L}_P^{D,\Box}$ cannot be dealt in our current setting, for $\Box$ is given its semantics in terms of conditional belief. A trivial result is that $\mathcal{L}_P^{C,D,\Box}$ is at least as expressive as both $\mathcal{L}_P^{C,D}$ and $\mathcal{L}_P^{C,\Box}$ as it subsumes both languages. Finally, assuming P is infinite it follows from Corollary 5.16 that $\mathcal{L}_P^{C,D}$ is more expressive than both $\mathcal{L}_P^C$ and $\mathcal{L}_P^D$. Given the, at least to us, surprising nature of the results established in this section, we shall not dare to venture a guess as to how these other combinations relate, leaving these questions for future work.

5.3 Conclusion and Discussion

In this chapter we showed that the notion of bisimulation developed in Chapter 4 was also the right fit for capturing modal equivalence of $\mathcal{L}_P^D$, that is, the extension of the epistemic language with a degrees of belief modality. Next, we proved several results on the expressive power of languages formed by different combinations of the conditional belief modality, the degrees of belief modality and the safe belief modality. We showed that under the assumption of having an infinite number of propositional symbols given, the conditional belief modality and degree of belief modality are able to express different properties of plausibility models, and as such stand as genuine doxastic alternatives. We furthermore showed that the addition of the safe belief modality to $\mathcal{L}_P^C$ yields a more expressive language.

That our notion of bisimulation developed in Chapter 4 is also appropriate for the language containing degrees of belief confirms our intuition that it is indeed the correct model-theoretic notion we have put forth. What is more, this implies that we can work with either or both modalities in a planning context, without increasing the size of contraction-minimal models nor the number of

non-bisimilar information cells. That we may want to do so is apparent from our expressivity results, for both modalities capture highly relevant aspects of planning with doxastic notions.

In terms of future work we have already accounted for the unresolved question of how the languages compare for a *finite* number of propositions, and furthermore noted languages that remains to be compared in terms of expressive power. Highly important to us is also the generalization of our notion of bisimulation to multi-agent plausibility models, which is current work. Our preliminary results show that there is a natural, albeit somewhat awkward, definition which corresponds to modal equivalence for the multi-agent epistemic-doxastic language containing the conditional belief modality. Such a result is vital for formalizing multi-agent planning with doxastic attitudes.

Chapter 6

Don't Plan for the Unexpected: Planning Based on Plausibility Models

This chapter is a replicate of [Andersen et al., 2014], accepted for publication in a special issue of Logique et Analyses devoted to the theme "Dynamics in Logic".

Don't Plan for the Unexpected: Planning Based on Plausibility Models

Mikkel Birkegaard Andersen Thomas Bolander
Martin Holm Jensen
DTU Compute, Technical University of Denmark

Abstract

We present a framework for automated planning based on plausibility models, as well as algorithms for computing plans in this framework. Our plausibility models include postconditions, as ontic effects are essential for most planning purposes. The framework presented extends a previously developed framework based on dynamic epistemic logic (DEL), without plausibilities/beliefs. In the pure epistemic framework, one can distinguish between strong and weak epistemic plans for achieving some, possibly epistemic, goal. By taking all possible outcomes of actions into account, a strong plan guarantees that the agent achieves this goal. Conversely, a weak plan promises only the possibility of leading to the goal. In real-life planning scenarios where the planning agent is faced with a high degree of uncertainty and an almost endless number of possible exogenous events, strong epistemic planning is not computationally feasible. Weak epistemic planning is not satisfactory either, as there is no way to qualify which of two weak plans is more likely to lead to the goal. This seriously limits the practical uses of weak planning, as the planning agent might for instance always choose a plan that relies on serendipity. In the present paper we introduce a planning framework with the potential of overcoming the problems of both weak and strong epistemic planning. This framework is based on

plausibility models, allowing us to define different types of plausibility planning. The simplest type of plausibility plan is one in which the goal will be achieved when all actions in the plan turn out to have the outcomes found most plausible by the agent. This covers many cases of everyday planning by human agents, where we—to limit our computational efforts—only plan for the most plausible outcomes of our actions.

6.1 Introduction

Whenever an agent deliberates about the future with the purpose of achieving a goal, she is engaging in the act of planning. Automated Planning is a widely studied area of AI dealing with such issues under many different assumptions and restrictions. In this paper we consider *planning under uncertainty* [Ghallab et al., 2004] (nondeterminism and partial observability), where the agent has knowledge and beliefs about the environment and how her actions affect it. We formulate scenarios using plausibility models obtained by merging the frameworks in [Baltag and Smets, 2006, van Ditmarsch and Kooi, 2008].

Example 6.1 (The Basement)**.** An agent is standing at the top of an unlit stairwell leading into her basement. If she walks down the steps in the dark, it's likely that she will trip. On the other hand, if the lights are on, she is certain to descend unharmed. There is a light switch just next to her, though she doesn't know whether the bulb is broken.

She wishes to find a plan that gets her safely to the bottom of the stairs. Planning in this scenario is contingent on the situation; e.g. is the bulb broken? Will she trip when attempting her descent? In planning terminology a plan that *might* achieve the goal is a *weak solution*, whereas one that *guarantees* it is a *strong solution*.

In this case, a weak solution is to simply descend the stairs in the dark, risking life and limb for a trip to the basement. On the other hand, there is no strong solution as the bulb might be broken (assuming it cannot be replaced). Intuitively, the *best* plan is to flick the switch (expecting the bulb to work) and then descend unharmed, something neither weak nor strong planning captures.

Extending the approach in [Andersen et al., 2012] to a logical framework incorporating beliefs via a plausibility ordering, we formalise plans which an agent considers most likely to achieve her goals. This notion is incorporated into algorithms developed for the framework in [Andersen et al., 2012], allowing us to

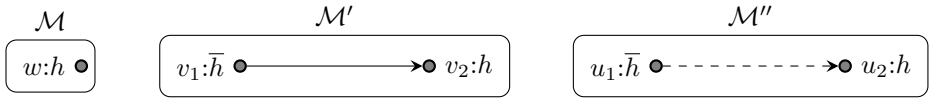


Figure 6.1: Three plausibility models.

synthesise plans like the *best* one in Example 6.1.

In the following section we present the logical framework we consider throughout the paper. Section 6.3 formalises planning in this framework, and introduces the novel concept of plausibility solutions to planning problems. As planning is concerned with representing possible ways in which the future can unfold, it turns out we need a belief modality corresponding to a globally connected plausibility ordering, raising some technical challenges. Section 6.4 introduces an algorithm for plan synthesis (i.e. generation of plans). Further we show that the algorithm is terminating, sound and complete. To prove termination, we must define bisimulations and bisimulation contractions.

6.2 Dynamic Logic of Doxastic Ontic Actions

The framework we need for planning is based on a dynamic logic of doxastic ontic actions. Actions can be epistemic (changing knowledge), doxastic (changing beliefs), ontic (changing facts) or any combination. The following formalisation builds on the *dynamic logic of doxastic actions* [Baltag and Smets, 2006], adding postconditions to event models as in [van Ditmarsch and Kooi, 2008]. We consider only the single-agent case. Before the formal definitions are given, we present some intuition behind the framework in the following example, which requires some familiarity with epistemic logic.

Example 6.2. Consider an agent and a coin biased towards heads, with the coin lying on a table showing heads (h). She contemplates tossing the coin and realizes that it can land either face up, but (due to nature of the coin) believes it will land heads up. In either case, after the toss she knows exactly which face is showing.

The initial situation is represented by the *plausibility model* (defined later) $\mathcal{M}$ and the contemplation by $\mathcal{M}''$ (see Figure 6.1). The two worlds u_1, u_2 are epistemically distinguishable ($u_1 \not\sim u_2$) and represent the observable nondeterministic outcome of the toss. The *dashed* directed edge signifies a *(global) plausibility relation*, where the direction indicates that she finds u_2 more plausible than u_1 (we overline proposition symbols that are false).

Example 6.3. Consider again the agent and biased coin. She now reasons about shuffling the coin under a dice cup, leaving the dice cup on top to conceal the coin. She cannot observe which face is up, but due to the bias of the coin believes it to be heads. She then reasons further about lifting the dice cup in this situation, and realises that she will observe which face is showing. Due to her beliefs about the shuffle she finds it most plausible that heads is observed.

The initial situation is again $\mathcal{M}$. Consider the model $\mathcal{M}'$, where the *solid* directed edge indicates a *local plausibility relation*, and the direction that v_2 is believed over v_1. By *local* we mean that the two worlds v_1, v_2 are *(epistemically) indistinguishable* ($v_1 \sim v_2$), implying that she is ignorant about whether h or $\neg h$ is the case.[1] Together this represents the concealed, biased coin. Her contemplations on lifting the cup is represented by the model $\mathcal{M}''$ as in the previous example.

In Example 6.2 the agent reasons about a non-deterministic action whose outcomes are *distinguishable* but not equally plausible, which is different from the initial contemplation in Example 6.3 where the outcomes are *not* distinguishable (due to the dice cup). In Example 3 she subsequently reasons about the observations made after a sensing action. In both examples she reasons about the future, and in both cases the final result is the model $\mathcal{M}''$. In Example 6.8 we formally elaborate on the actions used here.

It is the nature of the agent's ignorance that make $\mathcal{M}'$ and $\mathcal{M}''$ two inherently different situations. Whereas in the former she is ignorant about h due to the coin being concealed, her ignorance in the latter stems from not having lifted the cup yet. In general we can model ignorance either as a consequence of epistemic indistinguishability, or as a result of not yet having acted. Neither type subsumes the other and both are necessary for reasoning about actions. We capture this distinction by defining both local and global plausibility relations. The end result is that local plausibility talks about belief in a particular epistemic equivalence class, and global plausibility talks about belief in the entire model. We now remedy the informality we allowed ourselves so far by introducing the necessary definitions for a more formal treatment.

Definition 6.4 (Dynamic Language)**.** Let a countable set of propositional symbols P be given. The language $L(P)$ is given by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K\varphi \mid B^{\varphi}\varphi \mid X\varphi \mid [\mathcal{E}, e]\,\varphi$$

where $p \in P$, $\mathcal{E}$ is an *event model* on $L(P)$ as (simultaneously) defined below,

[1] In the remainder, we use (in)distinguishability without qualification to refer to epistemic (in)distinguishability.

and $e \in D(\mathcal{E})$. K is the *local* knowledge modality, B^{φ} the *global* conditional belief modality, X is a (non-standard) *localisation* modality (explained later) and $[\mathcal{E}, e]$ the dynamic modality.

We use the usual abbreviations for the other boolean connectives, as well as for the dual dynamic modality $\langle \mathcal{E}, e \rangle \varphi := \neg [\mathcal{E}, e] \neg \varphi$ and unconditional (or absolute) global belief $B\varphi := B^{\top}\varphi$. The duals of K and B^{φ} are denoted $\widehat{K}$ and $\widehat{B}^{\varphi}$.

$K\varphi$ reads as "the (planning) agent knows φ", $B^{\psi}\varphi$ as "conditional on ψ, the (planning) agent believes φ", and $[\mathcal{E}, e]\varphi$ as "after all possible executions of $(\mathcal{E}, e)$, φ holds". $X\varphi$ reads as "locally φ".

Definition 6.5 (Plausibility Models)**.** A *plausibility model* on a set of propositions P is a tuple $\mathcal{M} = (W, \sim, \leq, V)$, where

- W is a set of *worlds*,
- $\sim \subseteq W \times W$ is an equivalence relation called the *epistemic relation*,
- $\leq \subseteq W \times W$ is a connected well-preorder called the *plausibility relation*,[2]
- $V : P \to 2^W$ is a *valuation*.

$D(\mathcal{M}) = W$ denotes the *domain* of $\mathcal{M}$. For $w \in W$ we name $(\mathcal{M}, w)$ a *pointed plausibility model*, and refer to w as the *actual world* of $(\mathcal{M}, w)$. $<$ denotes the *strict plausibility relation*, that is $w < w'$ iff $w \leq w'$ and $w' \not\leq w$. $\simeq$ denotes *equiplausibility*, that is $w \simeq w'$ iff $w \leq w'$ and $w' \leq w$.

In our model illustrations a directed edge from w to w' indicates $w' \leq w$. By extension, strict plausibility is implied by unidirected edges and equiplausibility by bidirected edges. For the models in Figure 6.1, we have $v_1 \sim v_2$, $v_2 < v_1$ in $\mathcal{M}'$ and $u_1 \not\sim u_2$, $u_2 < u_1$ in $\mathcal{M}''$. The difference between these two models is in the epistemic relation, and is what gives rise to local (solid edges) and global (dashed edges) plausibility. In [Baltag and Smets, 2006] the local plausibility relation is defined as $\trianglelefteq := \sim \cap \leq$; i.e. $w \trianglelefteq w'$ iff $w \sim w'$ and $w \leq w'$. $\trianglelefteq$ is a *locally well-preordered relation*, meaning that it is a union of *mutually disjoint well-preorders*. Given a plausibility model, the domain of each element in this union corresponds to an $\sim$-equivalence class.

Our distinction between local and global is not unprecedented in the literature, but it can be a source of confusion. In [Baltag and Smets, 2006], $\leq$ was indeed

[2] A well-preorder is a reflexive, transitive binary relation s.t. every non-empty subset has minimal elements [Baltag and Smets, 2008].

connected (i.e. global), but in later versions of the framework [Baltag and Smets, 2008] this was no longer required. The iterative development in [van Ditmarsch, 2005] also discuss the distinction between local and global plausibility (named preference by the author). Relating the notions to the wording in [Baltag and Smets, 2006], $\leq$ captures *a priori* beliefs about *virtual* situations, *before* obtaining any direct information about the actual situation. On the other hand, $\trianglelefteq$ captures *a posteriori* beliefs about an *actual* situation, that is, the agent's beliefs *after* she obtains (or assumes) information about the actual world.

$\mathcal{M}''$ represents two distinguishable situations (v_1 and v_2) that are a result of reasoning about the future, with v_2 being considered more plausible than v_1. These situations are identified by restricting $\mathcal{M}''$ to its $\sim$-equivalence classes; i.e. $\mathcal{M}'' \restriction \{v_1\}$ and $\mathcal{M}'' \restriction \{v_2\}$. Formally, given an epistemic model $\mathcal{M}$, the *information cells* in $\mathcal{M}$ are the submodels of the form $\mathcal{M} \restriction [w]_\sim$ where $w \in D(\mathcal{M})$. We overload the term and name any $\sim$-connected plausibility model on P an *information cell*. This use is slightly different from the notion in [Baltag and Smets, 2008], where an information cell is an $\sim$-equivalence class rather than a restricted model. An immediate property of information cells is that $\leq = \trianglelefteq$; i.e. the local and global plausibility relations are identical. A partition of a plausibility model into its information cells corresponds to a *localisation* of the plausibility model, where each information cell represents a local situation. The (later defined) semantics of X enables reasoning about such localisations using formulas in the dynamic language.

Definition 6.6 (Event Models)**.** An *event model* on the language $L(P)$ is a tuple $\mathcal{E} = (E, \sim, \leq, pre, post)$, where

- E is a finite set of *(basic) events*,
- $\sim \subseteq E \times E$ is an equivalence relation called the *epistemic relation*,
- $\leq \subseteq E \times E$ is a connected well-preorder called the *plausibility relation*,
- $pre : E \to L(P)$ assigns to each event a *precondition*,
- $post : E \to (P \to L(P))$ assigns to each event a *postcondition* for each proposition. Each $post(e)$ is required to be only finitely different from the identity.

$D(\mathcal{E}) = E$ denotes the *domain* of $\mathcal{E}$. For $e \in E$ we name $(\mathcal{E}, e)$ a *pointed event model*, and refer to e as the *actual event* of $(\mathcal{E}, e)$. We use the same conventions for accessibility relations as in the case of plausibility models.

Definition 6.7 (Product Update)**.** Let $\mathcal{M} = (W, \sim, \leq, V)$ and $\mathcal{E} = (E, \sim', \leq', pre, post)$ be a plausibility model on P resp. event model on $L(P)$. The *product update* of $\mathcal{M}$ with $\mathcal{E}$ is the plausibility model denoted $\mathcal{M} \otimes \mathcal{E} = (W', \sim'', \leq'', V')$, where

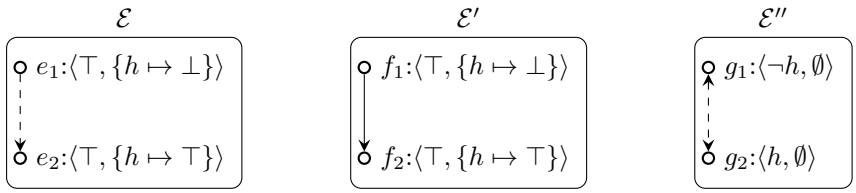


Figure 6.2: Three event models.

- $W' = \{(w, e) \in W \times E \mid \mathcal{M}, w \models pre(e)\}$,
- $\sim'' = \{((w, e), (v, f)) \in W' \times W' \mid w \sim v \text{ and } e \sim' f\}$,
- $\leq'' = \{((w, e), (v, f)) \in W' \times W' \mid e <' f \text{ or } (e \simeq' f \text{ and } w \leq v)\}$,
- $V'(p) = \{(w, e) \in W' \mid \mathcal{M}, w \models post(e)(p)\}$ for each $p \in P$.

The reader may consult [Baltag and Moss, 2004, Baltag and Smets, 2006, Baltag and Smets, 2008, van Ditmarsch and Kooi, 2008] for thorough motivations and explanations of the product update. Note that the event model's plausibilities take priority over those of the plausibility model (action-priority update).

Example 6.8. Consider Figure 6.2, where the event model $\mathcal{E}$ represents the biased non-deterministic coin toss of Example 6.2, $\mathcal{E}'$ shuffling the coin under a dice cup, and $\mathcal{E}''$ lifting the dice cup of Example 6.3. We indicate $\sim$ and $\leq$ with edges as in our illustrations of plausibility models. Further we use the convention of labelling basic events e by $\langle pre(e), post(e)\rangle$. We write $post(e)$ on the form $\{p_1 \mapsto \varphi_1, \ldots, p_n \mapsto \varphi_n\}$, meaning that $post(e)(p_i) = \varphi_i$ for all i, and $post(e)(q) = q$ for $q \notin \{p_1, \ldots, p_n\}$.

Returning to Example 6.2 we see that $\mathcal{M} \otimes \mathcal{E} = \mathcal{M}''$ where $u_1 = (w, e_1), u_2 = (w, e_2)$. In $\mathcal{E}$ we have that $e_2 < e_1$, which encodes the bias of the coin, and $e_1 \not\sim e_2$ encoding the observability, which leads to u_1 and u_2 being distinguishable.

Regarding Example 6.3 we have that $\mathcal{M} \otimes \mathcal{E}' = \mathcal{M}'$ (modulo renaming). In contrast to $\mathcal{E}$, we have that $f_1 \sim f_2$, representing the inability to see the face of the coin due to the dice cup. For the sensing action $\mathcal{E}''$, we have $\mathcal{M} \otimes \mathcal{E}' \otimes \mathcal{E}'' = \mathcal{M}''$, illustrating how, when events are equiplausible ($g_1 \simeq g_2$), the plausibilities of $\mathcal{M}'$ carry over to $\mathcal{M}''$.

We've shown examples of how the interplay between plausibility model and event model can encode changes in belief, and further how to model both ontic change and sensing. In [Bolander and Andersen, 2011] there is a more general treatment of action types, but here such a classification is not our objective.

Instead we simply encode actions as required for our exposition and leave these considerations as future work.

Among the possible worlds, $\leq$ gives an ordering defining what is believed. Given a plausibility model $\mathcal{M} = (W, \sim, \leq, V)$, any non-empty subset of W will have one or more minimal worlds with respect to $\leq$, since $\leq$ is a well-preorder. For $S \subseteq W$, the set of $\leq$-minimal worlds, denoted $Min_{\leq}S$, is defined as:

$$Min_{\leq}S = \{s \in S \mid \forall s' \in S : s \leq s'\}.$$

The worlds in $Min_{\leq}S$ are called the *most plausible* worlds in S. The worlds of $Min_{\leq}D(\mathcal{M})$ are referred to as the *most plausible* of $\mathcal{M}$. With belief defined via minimal worlds (see the definition below), the agent has the same beliefs for any $w \in D(\mathcal{M})$. Analogous to most plausible worlds, an information cell $\mathcal{M}'$ of $\mathcal{M}$ is called *most plausible* if $D(\mathcal{M}') \cap Min_{\leq}D(\mathcal{M}) \neq \emptyset$ ($\mathcal{M}'$ contains at least one of the most plausible worlds of $\mathcal{M}$).

Definition 6.9 (Satisfaction Relation)**.** Let a plausibility model $\mathcal{M} = (W, \sim, \leq, V)$ on P be given. The satisfaction relation is given by, for all $w \in W$:

$\mathcal{M}, w \models p$	iff $w \in V(p)$
$\mathcal{M}, w \models \neg\varphi$	iff *not* $\mathcal{M}, w \models \varphi$
$\mathcal{M}, w \models \varphi \wedge \psi$	iff $\mathcal{M}, w \models \varphi$ and $\mathcal{M}, w \models \psi$
$\mathcal{M}, w \models K\varphi$	iff $\mathcal{M}, v \models \varphi$ for all $w \sim v$
$\mathcal{M}, w \models B^{\psi}\varphi$	iff $\mathcal{M}, v \models \varphi$ for all $v \in Min_{\leq}\{u \in W \mid \mathcal{M}, u \models \psi\}$
$\mathcal{M}, w \models X\varphi$	iff $\mathcal{M} \restriction [w]_{\sim}, w \models \varphi$
$\mathcal{M}, w \models [\mathcal{E}, e]\,\varphi$	iff $\mathcal{M}, w \models pre(e)$ implies $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$

where $\varphi, \psi \in L(P)$ and $(\mathcal{E}, e)$ is a pointed event model. We write $\mathcal{M} \models \varphi$ to mean $\mathcal{M}, w \models \varphi$ for all $w \in D(\mathcal{M})$. Satisfaction of the dynamic modality for non-pointed event models $\mathcal{E}$ is introduced by abbreviation, viz. $[\mathcal{E}]\,\varphi := \bigwedge_{e \in \mathcal{D}(\mathcal{E})} [\mathcal{E}, e]\,\varphi$. Furthermore, $\langle \mathcal{E} \rangle\,\varphi := \neg\,[\mathcal{E}]\,\neg\varphi$.[3]

The reader may notice that the semantic clause for $\mathcal{M}, w \models X\varphi$ is equivalent to the clause for $\mathcal{M}, w \models [\mathcal{E}, e]\varphi$ when $[\mathcal{E}, e]$ is a public announcement of a *characteristic formula* [van Benthem, 1998] being true exactly at the worlds in $[w]_{\sim}$ (and any other world modally equivalent to one of these). In this sense, the X operator can be thought of as a public announcement operator, but a special one that always announces the current information cell. In the special case where $\mathcal{M}$ is an information cell, we have for all $w \in D(\mathcal{M})$ that $\mathcal{M}, w \models X\varphi$ iff $\mathcal{M}, w \models \varphi$.

[3] Hence, $\mathcal{M}, w \models \langle \mathcal{E} \rangle\,\varphi \Leftrightarrow \mathcal{M}, w \models \neg\,[\mathcal{E}]\,\neg\varphi \Leftrightarrow \mathcal{M}, w \models \neg(\bigwedge_{e \in \mathcal{D}(\mathcal{E})} [\mathcal{E}, e]\,\neg\varphi) \Leftrightarrow \mathcal{M}, w \models \bigvee_{e \in \mathcal{D}(\mathcal{E})} \neg\,[\mathcal{E}, e]\,\neg\varphi \Leftrightarrow \mathcal{M}, w \models \bigvee_{e \in D(\mathcal{E})} \langle \mathcal{E}, e \rangle\,\varphi$.

6.3 Plausibility Planning

The previous covered a framework for dealing with knowledge and belief in a dynamic setting. In the following, we will detail how a rational agent would adapt these concepts to model her own reasoning about how her actions affect the future. Specifically, we will show how an agent can predict whether or not a particular plan leads to a desired goal. This requires reasoning about the *conceivable* consequences of actions without *actually* performing them.

Two main concepts are required for our formulation of planning, both of which build on notions from the logic introduced in the previous section. One is that of states, a representation of the planning agent's view of the world at a particular time. Our states are plausibility models. The other concept is that of actions. These represent the agent's view of everything that can happen when she does something. Actions are event models, changing states into other states via product update.

In our case, the agent has knowledge and beliefs about the initial situation, knowledge and beliefs about actions, and therefore also knowledge and beliefs about the result of actions.

All of what follows regards planning in the internal perspective. Section 6.3.1 shows how plausibility models represent states, Section 6.3.2 how event models represent actions and Section 6.3.3 how these ideas can formalise planning problems with various kinds of solutions.

6.3.1 The Internal Perspective On States

In the internal perspective, an agent using plausibility models to represent her own view will, generally, not be able to point out the actual world. Consider again the model $\mathcal{M}$ in Figure 6.1, that has two indistinguishable worlds w_1 and w_2. If $\mathcal{M}$ is the agent's view of the situation, she will of course not be able to say which is the actual world. If she was, then the model *could not represent the situation where the two worlds are indistinguishable*. By requiring the agent to reason from non-pointed plausibility models only (a similar argument makes the case for non-pointed event models), we enforce the internal perspective.

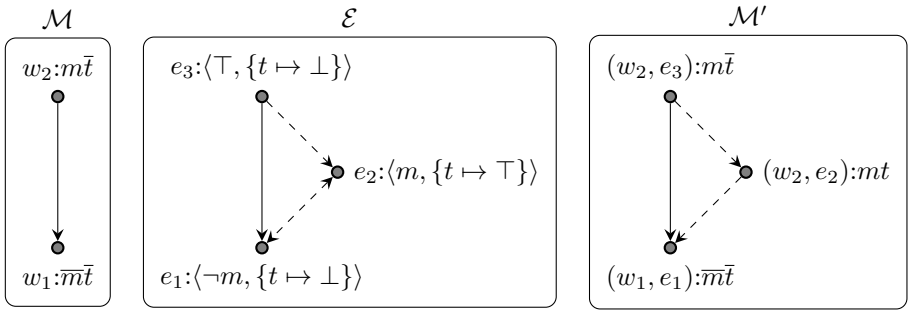


Figure 6.3: The situation before and after attempting to pay with a debit card, plus the event model depicting the attempt. This illustrates that the most plausible information cell can contain the least plausible world.

6.3.2 Reasoning About Actions

Example 6.10 (Friday Beer)**.** Nearing the end of the month, an agent is going to have an end-of-week beer with her coworkers. Wanting to save the cash she has on hand for the bus fare, she would like to buy the beer using her debit card. Though she isn't certain, she believes that there's no money ($\overline{m}$) on the associated account. Figure 6.3 shows this initial situation as $\mathcal{M}$, where $\bar{t}$ signifies that the transaction hasn't been completed. In this small example her goal is to make t true.

When attempting to complete the transaction (using a normal debit card reader), a number of different things can happen, captured by $\mathcal{E}$ in Figure 6.3. If there is money on the account, the transaction will go through (e_2), and if there isn't, it won't (e_1). This is how the card reader operates most of the time and why e_1 and e_2 are the most plausible events. Less plausible, but still possible, is that the reader malfunctions for some other reason (e_3). The only feedback the agent will receive is whether the transaction was completed, not the reasons why it did or didn't ($e_1 \sim e_3 \not\sim e_2$). That the agent finds out whether the transaction was successful is why we do not collapse e_1 and e_2 to one event e' with $pre(e') = \top$ and $post(e')(t) = m$.

$\mathcal{M} \otimes \mathcal{E}$ expresses the agent's view on the possible outcomes of attempting the transaction. The model $\mathcal{M}'$ is the bisimulation contraction of $\mathcal{M} \otimes \mathcal{E}$, according to the definition in Section 6.4.1 (the world (w_1, e_3) having been removed, as it is bisimilar to (w_1, e_1)).

$\mathcal{M}'$ consists of two information cells, corresponding to whether or not the transaction was successful. What she believes will happen is given by the global

plausibility relation. When actually attempting the transaction the result will be one of the information cells of $\mathcal{M}'$, namely $\mathcal{M}_{\bar{t}} = \mathcal{M}' \restriction \{(w_1, e_1), (w_2, e_3)\}$ or $\mathcal{M}_t = \mathcal{M}' \restriction \{(w_2, e_2)\}$, in which she will know $\neg t$ and t respectively. As (w_1, e_1) is the most plausible, we can say that she *expects* to end up in (w_1, e_1), and, by extension, in the information cell $\mathcal{M}_{\bar{t}}$: She expects to end up in a situation where she knows $\neg t$, but is ignorant concerning m. If, unexpectedly, the transaction *is* successful, she will know that the balance is sufficient (m). The most plausible information cell(s) in a model are those the agent expects. That (w_2, e_3) is in the expected information cell, when the globally more plausible world (w_2, e_2) is not, might seem odd. It isn't. The partitioning of $\mathcal{M}$ into the information cells $\mathcal{M}_{\bar{t}}$ and $\mathcal{M}_t$ suggests that she will sense the value of t ($\neg t$ holds everywhere in the former, t everywhere in the latter). As she expects to find out that t does not to hold, she expects to be able to rule out all the worlds in which t *does* hold. Therefore, she expects to be able to rule out (w_2, e_2) and *not* (w_2, e_3) (or w_1, e_1). This gives $\mathcal{M}' \models BX(K\neg t \wedge B\neg m \wedge \widehat{K}m)$: She expects to come to know that the transaction has failed and that she will believe there's no money on the account (though she does consider it possible that there is).

Under the definition of planning that is to follow in Section 6.3.3, an agent has a number of actions available to construct plans. She needs a notion of which actions can be considered at different stages of the planning process. As in the planning literature, we call this notion *applicability*.

Definition 6.11 (Applicability)**.** An event model $\mathcal{E}$ is said to be *applicable* in a plausibility model $\mathcal{M}$ if $\mathcal{M} \models \langle\mathcal{E}\rangle \top$.

Unfolding the definition of $\langle\mathcal{E}\rangle$, we see what applicability means:

$$
\begin{aligned}
&\mathcal{M} \models \langle\mathcal{E}\rangle \top \Leftrightarrow \forall w \in D(\mathcal{M}) : \mathcal{M}, w \models \langle\mathcal{E}\rangle \top \Leftrightarrow \\
&\forall w \in D(\mathcal{M}) : \mathcal{M}, w \models \vee_{e \in D(\mathcal{E})} \langle\mathcal{E}, e\rangle \top \Leftrightarrow \\
&\forall w \in D(\mathcal{M}), \exists e \in D(\mathcal{E}) : \mathcal{M}, w \models \langle\mathcal{E}, e\rangle \top \Leftrightarrow \\
&\forall w \in D(\mathcal{M}), \exists e \in D(\mathcal{E}) : \mathcal{M}, w \models pre(e) \text{ and } \mathcal{M} \otimes \mathcal{E}, (w, e) \models \top \Leftrightarrow \\
&\forall w \in D(\mathcal{M}), \exists e \in D(\mathcal{E}) : \mathcal{M}, w \models pre(e).
\end{aligned}
$$

This says that no matter which is the actual world (it must be one of those considered possible), the action defines an outcome. This concept of applicability is equivalent to the one in [Bolander and Andersen, 2011]. The discussion in [de Lima, 2007, sect. 6.6] also notes this aspect, insisting that actions must be *meaningful*. The same sentiment is expressed by our notion of applicability.

Proposition 6.12. *Given a plausibility model $\mathcal{M}$ and an applicable event model $\mathcal{E}$, we have $D(\mathcal{M} \otimes \mathcal{E}) \neq \emptyset$.*

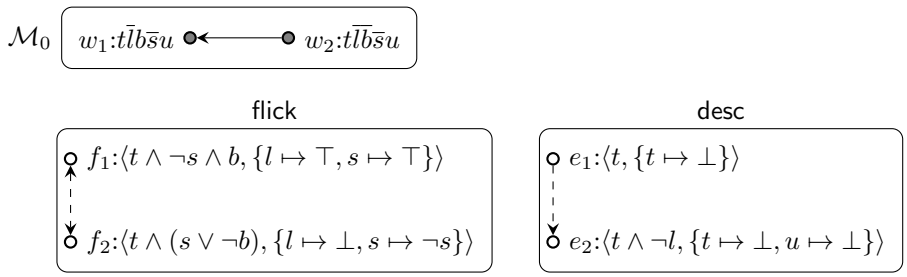


Figure 6.4: An information cell, $\mathcal{M}_0$, and two event models, flick and desc.

The product update $\mathcal{M} \otimes \mathcal{E}$ expresses the outcome(s) of doing $\mathcal{E}$ in the situation $\mathcal{M}$, in the planning literature called *applying* $\mathcal{E}$ in $\mathcal{M}$. The dynamic modality $[\mathcal{E}]$ expresses reasoning about what holds after applying $\mathcal{E}$.

Lemma 6.13. *Let $\mathcal{M}$ be a plausibility model and $\mathcal{E}$ an event model. Then $\mathcal{M} \models [\mathcal{E}]\varphi$ iff $\mathcal{M} \otimes \mathcal{E} \models \varphi$.*

Proof. $\mathcal{M} \models [\mathcal{E}]\varphi \Leftrightarrow \forall w \in \mathcal{D}(\mathcal{M}) : \mathcal{M}, w \models [\mathcal{E}]\varphi \Leftrightarrow$
$\forall w \in \mathcal{D}(\mathcal{M}) : \mathcal{M}, w \models \bigwedge_{e \in \mathcal{D}(\mathcal{E})} [\mathcal{E}, e]\varphi \Leftrightarrow \forall (w, e) \in \mathcal{D}(\mathcal{M}) \times \mathcal{D}(\mathcal{E}) : \mathcal{M}, w \models [\mathcal{E}, e]\varphi \Leftrightarrow$
$\forall (w, e) \in \mathcal{D}(\mathcal{M}) \times \mathcal{D}(\mathcal{E}) : \mathcal{M}, w \models pre(e)$ implies $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi \Leftrightarrow$
$\forall (w, e) \in D(\mathcal{M} \otimes \mathcal{E}) : \mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi \Leftrightarrow \mathcal{M} \otimes \mathcal{E} \models \varphi.$ □

Here we are looking at *global satisfaction*, by evaluating $[\mathcal{E}]\varphi$ in all of $\mathcal{M}$, rather than a specific world. The reason is that evaluation in planning must happen from the perspective of the planning agent and its "information state". Though one of the worlds of $\mathcal{M}$ is the actual world, the planning agent is ignorant about which it is. Whatever plan it comes up with, it must work in all of the worlds which are indistinguishable to the agent, that is, in the entire model. A similar point, and a similar solution, is found in [Jamroga and Ågotnes, 2007].

Example 6.14. We now return to the agent from Example 6.1. Her view of the initial situation ($\mathcal{M}_0$) and her available actions (flick and desc) are seen in Figure 6.4. The propositional letters mean t: "top of stairs", l: "light on", b: "bulb working", s : "switch on" and u: "unharmed". Initially, in $\mathcal{M}_0$, she believes that the bulb is working, and knows that she is at the top of the stairs, unharmed and that the switch and light is off: $\mathcal{M}_0 \models Bb \wedge K(t \wedge u \wedge \neg l \wedge \neg s)$.

flick and desc represent flicking the light switch and trying to descend the stairs, respectively. Both require being at the top of the stairs (t). f_1 of flick expresses that if the bulb is working, turning on the switch will turn on the light, and f_2

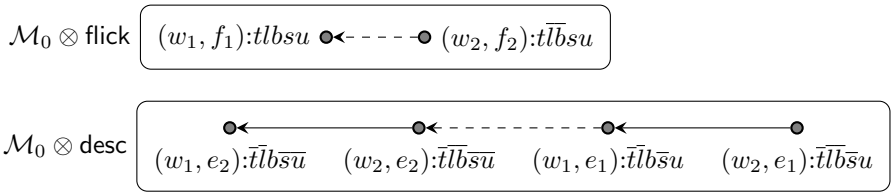


Figure 6.5: The models resulting from applying the actions flick and desc in $\mathcal{M}_0$. Reflexive edges are not shown and the transitive closure is left implicit.

that if the bulb is broken or the switch is currently on, the light will be off. The events are epistemically distinguishable, as the agent will be able to tell whether the light is on or off. desc describes descending the stairs, with or without the light on. e_1 covers the agent descending the stairs unharmed, and can happen regardless of there being light or not. The more plausible event e_2 represents the agent stumbling, though this can only happen in the dark. If the light is on, she will descend safely. Definition 6.11 and Lemma 6.13 let us express the action sequences possible in this scenario.

- $\mathcal{M}_0 \models \langle \mathsf{flick} \rangle \top \wedge \langle \mathsf{desc} \rangle \top$. The agent can initially do either flick or desc.
- $\mathcal{M}_0 \models [\mathsf{flick}] \langle \mathsf{desc} \rangle \top$. After doing flick, she can do desc.
- $\mathcal{M}_0 \models [\mathsf{desc}] (\neg \langle \mathsf{flick} \rangle \top \wedge \neg \langle \mathsf{desc} \rangle \top)$. Nothing can be done after desc.

Figure 6.5 shows the plausibility models arising from doing flick and desc in $\mathcal{M}_0$. Via Lemma 6.13 she can now conclude:

- $\mathcal{M}_0 \models [\mathsf{flick}] (Kb \vee K\neg b)$: Flicking the light switch gives knowledge of whether the bulb works or not.
- $\mathcal{M}_0 \models [\mathsf{flick}] BKb$. She expects to come to know that it works.
- $\mathcal{M}_0 \models [\mathsf{desc}] (K\neg t \wedge B\neg u)$. Descending the stairs in the dark will definitely get her to the bottom, though she believes she will end up hurting herself.

6.3.3 Planning

We now turn to formalising planning and then proceed to answer two questions of particular interest: How do we verify that a given plan achieves a goal? And can we compute such plans? This section deals with the first question, plan verification, while the second, plan synthesis, is detailed in Section 6.4.

Definition 6.15 (Plan Language)**.** Given a finite set A of event models on $L(P)$, the *plan language* $\mathcal{L}(P, \mathsf{A})$ is given by:

$$\pi ::= \mathcal{E} \mid \mathsf{skip} \mid \mathsf{if}\ \varphi\ \mathsf{then}\ \pi\ \mathsf{else}\ \pi \mid \pi; \pi$$

where $\mathcal{E} \in \mathsf{A}$ and $\varphi \in L(P)$. We name members π of this language *plans*, and use $\mathsf{if}\ \varphi\ \mathsf{then}\ \pi$ as shorthand for $\mathsf{if}\ \varphi\ \mathsf{then}\ \pi\ \mathsf{else}\ \mathsf{skip}$.

The reading of the plan constructs are "do $\mathcal{E}$", "do nothing", "if φ then π, else π'", and "first π then π'" respectively. In the translations provided in Definition 6.16, the condition of the if-then-else construct becomes a K-formula, ensuring that branching depends only on worlds which are distinguishable to the agent. The idea is similar to the *meaningful plans* of [de Lima, 2007], where branching is allowed on *epistemically interpretable formulas* only.

Definition 6.16 (Translation)**.** Let α be one of s, w, sp or wp. We define an *α-translation* as a function $[\cdot]_\alpha : \mathcal{L}(P, \mathsf{A}) \to (L(P) \to L(P))$:

$$[\mathcal{E}]_\alpha \varphi := \langle \mathcal{E} \rangle \top \wedge \begin{cases} [\mathcal{E}]\, XK\varphi & \text{if } \alpha = s \\ \widehat{K}\, \langle \mathcal{E} \rangle\, XK\varphi & \text{if } \alpha = w \\ [\mathcal{E}]\, BXK\varphi & \text{if } \alpha = sp \\ [\mathcal{E}]\, \widehat{B}XK\varphi & \text{if } \alpha = wp \end{cases}$$

$$[\mathsf{skip}]_\alpha \varphi := \varphi$$

$$[\mathsf{if}\ \varphi'\ \mathsf{then}\ \pi\ \mathsf{else}\ \pi']_\alpha \varphi := (K\varphi' \to [\pi]_\alpha \varphi) \wedge (\neg K\varphi' \to [\pi']_\alpha \varphi)$$

$$[\pi; \pi']_\alpha \varphi := [\pi]_\alpha ([\pi']_\alpha \varphi)$$

We call $[\cdot]_s$ the *strong translation*, $[\cdot]_w$ the *weak translation*, $[\cdot]_{sp}$ the *strong plausibility* translation and $[\cdot]_{wp}$ the *weak plausibility* translation.

The translations are constructed specifically to make the following lemma hold, providing a semantic interpretation of plans (leaving out skip and $\pi_1; \pi_2$).

Lemma 6.17. *Let $\mathcal{M}$ be an information cell, $\mathcal{E}$ an event model and φ a formula of $L(P)$. Then:*

1. $\mathcal{M} \models [\mathcal{E}]_s \varphi$ iff $\mathcal{M} \models \langle \mathcal{E} \rangle \top$ *and for each information cell* $\mathcal{M}'$ *of* $\mathcal{M} \otimes \mathcal{E}$: $\mathcal{M}' \models \varphi$.
2. $\mathcal{M} \models [\mathcal{E}]_w \varphi$ iff $\mathcal{M} \models \langle \mathcal{E} \rangle \top$ *and for some information cell* $\mathcal{M}'$ *of* $\mathcal{M} \otimes \mathcal{E}$: $\mathcal{M}' \models \varphi$.

3. $\mathcal{M} \models [\mathcal{E}]_{sp}\varphi$ iff $\mathcal{M} \models \langle\mathcal{E}\rangle\top$ *and for each most plausible information cell* $\mathcal{M}'$ *of* $\mathcal{M} \otimes \mathcal{E} : \mathcal{M}' \models \varphi$.
4. $\mathcal{M} \models [\mathcal{E}]_{wp}\varphi$ iff $\mathcal{M} \models \langle\mathcal{E}\rangle\top$ *and for some most plausible information cell* $\mathcal{M}'$ *of* $\mathcal{M} \otimes \mathcal{E} : \mathcal{M}' \models \varphi$.
5. $\mathcal{M} \models [\mathsf{if}\ \varphi'\ \mathsf{then}\ \pi\ \mathsf{else}\ \pi']_\alpha\varphi$ iff
   $(\mathcal{M} \models \varphi'$ *implies* $\mathcal{M} \models [\pi]_\alpha\varphi)$ *and* $(\mathcal{M} \not\models \varphi'$ *implies* $\mathcal{M} \models [\pi']_\alpha\varphi)$.

Proof. We only prove 4 and 5, as 1–4 are very similar. For 4 we have:

$\mathcal{M} \models [\mathcal{E}]_{wp}\,\varphi \;\Leftrightarrow\; \mathcal{M} \models \langle\mathcal{E}\rangle\top \wedge [\mathcal{E}]\,\widehat{B}XK\varphi \;\Leftrightarrow^{\text{Lemma 6.13}}$
$\mathcal{M} \models \langle\mathcal{E}\rangle\top$ and $\mathcal{M} \otimes \mathcal{E} \models \widehat{B}XK\varphi \;\Leftrightarrow$
$\mathcal{M} \models \langle\mathcal{E}\rangle\top$ and $\forall(w,e) \in D(\mathcal{M} \otimes \mathcal{E}) : \mathcal{M} \otimes \mathcal{E}, (w,e) \models \widehat{B}XK\varphi \;\Leftrightarrow^{\text{Prop. 6.12}}$
$\mathcal{M} \models \langle\mathcal{E}\rangle\top$ and $\exists(w,e) \in Min_{\leq}D(\mathcal{M} \otimes \mathcal{E}) : \mathcal{M} \otimes \mathcal{E}, (w,e) \models XK\varphi \;\Leftrightarrow$
$\mathcal{M} \models \langle\mathcal{E}\rangle\top$ and $\exists(w,e) \in Min_{\leq}D(\mathcal{M} \otimes \mathcal{E}) : \mathcal{M} \otimes \mathcal{E} \upharpoonright [(w,e)]_\sim, (w,e) \models K\varphi \;\Leftrightarrow$
$\mathcal{M} \models \langle\mathcal{E}\rangle\top$ and $\exists(w,e) \in Min_{\leq}D(\mathcal{M} \otimes \mathcal{E}) : \mathcal{M} \otimes \mathcal{E} \upharpoonright [(w,e)]_\sim \models \varphi \;\Leftrightarrow$
$\mathcal{M} \models \langle\mathcal{E}\rangle\top$ and in some most plausible information cell $\mathcal{M}'$ of $\mathcal{M} \otimes \mathcal{E}$, $\mathcal{M}' \models \varphi$.

For if-then-else, first note that:

$\mathcal{M} \models \neg K\varphi' \to [\pi]_\alpha\varphi \;\Leftrightarrow\; \forall w \in D(\mathcal{M})\colon \mathcal{M}, w \models \neg K\varphi' \to [\pi]_\alpha\varphi \;\Leftrightarrow$
$\forall w \in D(\mathcal{M})\colon \mathcal{M}, w \models \neg K\varphi'$ implies $\mathcal{M}, w \models [\pi]_\alpha\varphi \;\Leftrightarrow^{\mathcal{M}\text{ is an info. cell}}$
$\forall w \in D(\mathcal{M})\colon$ if $\mathcal{M}, v \models \neg\varphi'$ for some $v \in D(\mathcal{M})$ then $\mathcal{M}, w \models [\pi]_\alpha\varphi \;\Leftrightarrow$
if $\mathcal{M}, v \models \neg\varphi'$ for some $v \in D(\mathcal{M})$ then $\forall w \in D(\mathcal{M})\colon \mathcal{M}, w \models [\pi]_\alpha\varphi \;\Leftrightarrow$
$\mathcal{M} \not\models \varphi'$ implies $\mathcal{M} \models [\pi']_\alpha\varphi$.

Similarly, we can prove:

$$\mathcal{M} \models K\varphi' \to [\pi]_\alpha\varphi \;\Leftrightarrow\; \mathcal{M} \models K\varphi' \text{ implies } \mathcal{M} \models [\pi']_\alpha\varphi.$$

Using these facts, we get:

$\mathcal{M} \models [\mathsf{if}\ \varphi'\ \mathsf{then}\ \pi\ \mathsf{else}\ \pi']_\alpha\varphi \;\Leftrightarrow\; \mathcal{M} \models (K\varphi' \to [\pi]_\alpha\varphi) \wedge (\neg K\varphi' \to [\pi']_\alpha\varphi) \;\Leftrightarrow$
$\mathcal{M} \models K\varphi' \to [\pi]_\alpha\varphi$ and $\mathcal{M} \models \neg K\varphi' \to [\pi']_\alpha\varphi \;\Leftrightarrow$
$(\mathcal{M} \models \varphi'$ implies $\mathcal{M} \models [\pi]_\alpha\varphi)$ and $(\mathcal{M} \not\models \varphi'$ implies $\mathcal{M} \models [\pi']_\alpha\varphi)$.

□

Using XK (as is done in all translations) means that reasoning after an action is relative to a particular information cell (as $\mathcal{M}, w \models XK\varphi \Leftrightarrow \mathcal{M} \upharpoonright [w]_\sim, w \models K\varphi \Leftrightarrow \mathcal{M} \upharpoonright [w]_\sim \models \varphi$).

Definition 6.18 (Planning Problems and Solutions)**.** Let P be a finite set of propositional symbols. A planning problem on P is a triple $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$ where

- $\mathcal{M}_0$ is a finite information cell on P called the *initial state*.
- A is a finite set of event models on $L(P)$ called the *action library*.
- $\varphi_g \in L(P)$ is the *goal (formula)*.

A plan $\pi \in \mathcal{L}(P, \mathsf{A})$ is an α-*solution* to $\mathcal{P}$ if $\mathcal{M}_0 \models [\pi]_\alpha \varphi_g$. For a specific choice of $\alpha = s/w/sp/wp$, we will call π a *strong/weak/strong plausibility/weak plausibility*-solution respectively.

Given a π, we wish to check whether π is an α-solution (for some particular α) to $\mathcal{P}$. This can be done via model checking the dynamic formula given by the translation $[\pi]_\alpha \varphi_g$ in the initial state of $\mathcal{P}$.

A strong solution π is one that guarantees that φ_g will hold after executing it ("π achieves φ_g"). If π is a weak solution, it achieves φ_g for at least one particular sequence of outcomes. Strong and weak plausibility-solutions are as strong- and weak-solutions, except that they need only achieve φ_g for *all of/some of* the most plausible outcomes.

Example 6.19. The basement scenario (Example 6.1) can be formalised as the planning problem $\mathcal{P}_B = (\mathcal{M}_0, \{\mathsf{flick}, \mathsf{desc}\}, \varphi_g)$ with $\mathcal{M}_0$, flick and desc being defined in Figure 6.4 and $\varphi_g = \neg t \wedge u$. Let $\pi_1 = \mathsf{desc}$. We then have that:

$$\mathcal{M}_0 \models [\mathsf{desc}]_w (\neg t \wedge u) \Leftrightarrow \mathcal{M}_0 \models \langle \mathsf{desc} \rangle \top \wedge \widehat{K} \langle \mathsf{desc} \rangle XK(\neg t \wedge u) \Leftrightarrow^{\mathsf{desc} \text{ is applic.}}$$
$$\mathcal{M}_0 \models \widehat{K} \langle \mathsf{desc} \rangle XK(\neg t \wedge u) \Leftrightarrow \exists w \in D(\mathcal{M}_0) : \mathcal{M}_0, w \models \langle \mathsf{desc} \rangle XK(\neg t \wedge u).$$

Picking w_1, we have

$$\mathcal{M}_0, w_1 \models \langle \mathsf{desc} \rangle XK(\neg t \wedge u) \Leftrightarrow \mathcal{M}_0 \otimes \mathsf{desc}, (w_1, e_1) \models XK(\neg t \wedge u) \Leftrightarrow$$
$$\mathcal{M}_0 \otimes \mathsf{desc} \upharpoonright [(w_1, e_1)]_\sim \models (\neg t \wedge u)$$

which holds as seen in Figure 6.5. Thus, π_1 is a weak solution. Further, Lemma 6.17 tells us that π_1 is not a $s/wp/sp$ solution, as u does not hold in the (most plausible) information cell $\mathcal{M} \otimes \mathsf{desc} \upharpoonright \{(w_1, e_2), (w_2, e_2)\}$.

The plan $\pi_2 = \mathsf{flick}; \mathsf{desc}$ is a strong plausibility solution, as can be verified by $\mathcal{M}_0 \models [\pi_2]_{sp} (\neg t \wedge u)$. Without an action for replacing the lightbulb, there are no strong solutions. Let $\mathsf{replace}$ be the action in Figure 6.6, where $post(r_1)(u) = \neg s$

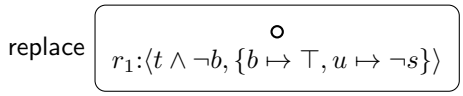


Figure 6.6: Event model for replacing a broken bulb.

signifies that if the power is on, the agent will hurt herself, and define a new problem $\mathcal{P}'_B = \{\mathcal{M}_0, \{\mathsf{flick}, \mathsf{desc}, \mathsf{replace}\}, \varphi_g)$. Then

$$\pi_3 = \mathsf{flick}; (\mathsf{if}\ \neg l\ \mathsf{then}\ \mathsf{flick}; \mathsf{replace}; \mathsf{flick}); \mathsf{desc}$$

is a strong solution (we leave verification to the reader): If the light comes on after flicking the switch (as expected) she can safely walk down the stairs. If it does not, she turns off the power, replaces the broken bulb, turns the power on again (this time knowing that the light will come on), and then proceeds as before.

Besides being an *sp*-solution, π_2 is also a *w*- and a *wp*-solution, indicating a hierarchy of strengths of solutions. This should come as no surprise, given both the formal and intuitive meaning of planning and actions presented so far. In fact, this hierarchy exists for any planning problem, as shown by the following result which is a consequence of Lemma 6.17 (stated without proof).

Lemma 6.20. *Let $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$ be a planning problem. Then:*

- *Any strong solution to $\mathcal{P}$ is also a strong plausibility solution:* $\mathcal{M}_0 \models [\pi]_s \varphi_g \Rightarrow \mathcal{M}_0 \models [\pi]_{sp} \varphi_g$.
- *Any strong plausibility solution to $\mathcal{P}$ is also a weak plausibility solution:* $\mathcal{M}_0 \models [\pi]_{sp} \varphi_g \Rightarrow \mathcal{M}_0 \models [\pi]_{wp} \varphi_g$.
- *Any weak plausibility solution to $\mathcal{P}$ is also a weak solution:* $\mathcal{M}_0 \models [\pi]_{wp} \varphi_g \Rightarrow \mathcal{M}_0 \models [\pi]_{w} \varphi_g$.

6.4 Plan Synthesis

In this section we show how to synthesise conditional plans for solving planning problems. Before we can give the concrete algorithms, we establish some technical results which are stepping stones to proving termination of our planning algorithm, and hence decidability of plan existence in our framework.

6.4.1 Bisimulations, contractions and modal equivalence

We now define bisimulations on plausibility models. For our purpose it is sufficient to define bisimulations on $\sim$-connected models, that is, on information cells.[4] First we define a *normal plausibility relation* which will form the basis of our bisimulation definition.

Definition 6.21 (Normality)**.** Given is an information cell $\mathcal{M} = (W, \sim, \leq, V)$ on P. By slight abuse of language, two worlds $w, w' \in W$ are said to *have the same valuation* if for all $p \in P$: $w \in V(p) \Leftrightarrow w' \in V(p)$. Define an equivalence relation on W: $w \approx w'$ iff w and w' has the same valuation. Now define $w \preceq w'$ iff $Min_{\leq}([w]_{\approx}) \leq Min_{\leq}([w']_{\approx})$. This defines the *normal plausibility relation*. $\mathcal{M}$ is called *normal* if $\preceq \; = \; \leq$. The *normalisation* of $\mathcal{M} = (W, \sim, \leq, V)$ is $\mathcal{M}' = (W, \sim, \preceq, V)$.

Definition 6.22 (Bisimulation)**.** Let $\mathcal{M} = (W, \sim, \leq, V)$ and $\mathcal{M}' = (W', \sim', \leq', V')$ be information cells on P. A non-empty relation $\mathcal{R} \subseteq W \times W'$ is a *bisimulation* between $\mathcal{M}$ and $\mathcal{M}'$ (and $\mathcal{M}, \mathcal{M}'$ are called *bisimilar*) if for all $(w, w') \in \mathcal{R}$:

[atom] For all $p \in P$: $w \in V(p)$ iff $w' \in V'(p)$.

[forth] If $v \in W$ and $v \preceq w$ then there is a $v' \in W'$ s.t. $v' \preceq' w'$ and $(v, v') \in \mathcal{R}$.

[back] If $v' \in W'$ and $v' \preceq w'$ then there is a $v \in W$ s.t. $v \preceq w$ and $(v, v') \in \mathcal{R}$.

If $\mathcal{R}$ has domain W and codomain W', it is called *total*. If $\mathcal{M} = \mathcal{M}'$, it is called an *autobisimulation* (on $\mathcal{M}$). Two worlds w and w' of an information cell $\mathcal{M} = (W, \sim, \leq, V)$ are called *bisimilar* if there exists an autobisimulation $\mathcal{R}$ on $\mathcal{M}$ with $(w, w') \in \mathcal{R}$.

We are here only interested in total bisimulations, so, unless otherwise stated, we assume this in the following. Note that our definition of bisimulation immediately implies that there exists a (total) bisimulation between any information cell and its normalisation. Note also that for normal models, the bisimulation definition becomes the standard modal logic one.[5]

[4] The proper notion of bisimulation for plausibility structures is explored in more detail by Andersen, Bolander, van Ditmarsch and Jensen in ongoing research. A similar notion for slightly different types of plausibility structures is given in [van Ditmarsch, pear]. Surprisingly, Demey does not consider our notion of bisimulation in his thorough survey [Demey, 2011] on different notions of bisimulation for plausibility structures.

[5] We didn't include a condition for the epistemic relation, $\sim$, in [back] and [forth], simply because we are here only concerned with $\sim$-connected models.

Lemma 6.23. *If two worlds of an information cell have the same valuation they are bisimilar.*

Proof. Assume worlds w and w' of an information cell $\mathcal{M} = (W, \sim, \leq, V)$ have the same valuation. Let $\mathcal{R}$ be the relation that relates each world of $\mathcal{M}$ to itself and additionally relates w to w'. We want to show that $\mathcal{R}$ is a bisimulation. This amounts to showing [atom], [forth] and [back] for the pair $(w, w') \in \mathcal{R}$. [atom] holds trivially since $w \approx w'$. For [forth], assume $v \in W$ and $v \preceq w$. We need to find a $v' \in W$ s.t. $v' \preceq w'$ and $(v, v') \in \mathcal{R}$. Letting $v' = v$, it suffices to prove $v \preceq w'$. Since $w \approx w'$ this is immediate: $v \preceq w \Leftrightarrow Min_{\leq}([v]_{\approx}) \leq Min_{\leq}([w]_{\approx}) \overset{w \approx w'}{\Leftrightarrow} Min_{\leq}([v]_{\approx}) \leq Min_{\leq}([w']_{\approx}) \Leftrightarrow v \preceq w'$. [back] is proved similarly. □

Unions of autobisimulations are autobisimulations. We can then in the standard way define the *(bisimulation) contraction* of a normal information cell as its quotient with respect to the union of all autobisimulations [Blackburn and van Benthem, 2006].[6] The contraction of a non-normal model is taken to be the contraction of its normalisation. In a contracted model, no two worlds are bisimilar, by construction. Hence, by Lemma 6.23, no two worlds have the same valuation. Thus, the contraction of an information cell on a finite set of proposition symbols P contains at most $2^{|P|}$ worlds. Since any information cell is bisimilar to its contraction [Blackburn and van Benthem, 2006], this shows that there can only exist finitely many non-bisimilar information cells on any given finite set P.

Two information cells $\mathcal{M}$ and $\mathcal{M}'$ are called *modally equivalent*, written $\mathcal{M} \equiv \mathcal{M}'$, if for all formulas φ in $L(P)$: $\mathcal{M} \models \varphi \Leftrightarrow \mathcal{M}' \models \varphi$. Otherwise, they are called *modally inequivalent*. We now have the following standard result (the result is standard for standard modal languages and bisimulations, but it is not trivial that it also holds here).

Theorem 6.24. *If two information cells are (totally) bisimilar they are modally equivalent.*

Proof. We need to show that if $\mathcal{R}$ is a total bisimulation between information cells $\mathcal{M}$ and $\mathcal{M}'$, then for all formulas φ of $L(P)$: $\mathcal{M} \models \varphi \Leftrightarrow \mathcal{M}' \models \varphi$. First we show that we only have to consider formulas φ of the static sublanguage of $L(P)$, that is, the language without the $[\mathcal{E}, e]$ modalities. In [Baltag and Smets,

[6] More precisely, let $\mathcal{M}$ be a normal information cell and let $\mathcal{R}$ be the union of all autobisimulations on $\mathcal{M}$. Then the contraction $\mathcal{M}' = (W', \sim', \leq', V')$ of $\mathcal{M}$ has as worlds the equivalence classes $[w]_{\mathcal{R}} = \{w' \mid (w, w') \in \mathcal{R}\}$ and has $[w]_{\mathcal{R}} \leq' [w']_{\mathcal{R}}$ iff $v \leq v'$ for some $v \in [w]_{\mathcal{R}}$ and $v' \in [w']_{\mathcal{R}}$.

2006], reduction axioms from the dynamic to the static language are given for a language similar to $L(P)$. The differences in language are our addition of postconditions and the fact that our belief modality is defined from the global plausibility relation rather than being localised to epistemic equivalence classes. The latter difference is irrelevant when only considering information cells as we do here. The former difference of course means that the reduction axioms presented in [Baltag and Smets, 2006] will not suffice for our purpose. [van Ditmarsch and Kooi, 2008] shows that adding postconditions to the language without the doxastic modalities only requires changing the reduction axiom for $[\mathcal{E}, e]\, p$, where p is a propositional symbol. Thus, if we take the reduction axioms of [Baltag and Smets, 2006] and replace the reduction axiom for $[\mathcal{E}, e]\, p$ by the one in [van Ditmarsch and Kooi, 2008], we get reduction axioms for our framework. We leave out the details.

We now need to show that if $\mathcal{R}$ is a total bisimulation between information cells $\mathcal{M}$ and $\mathcal{M}'$, then for all $[\mathcal{E}, e]$-free formulas φ of $L(P)$: $\mathcal{M} \models \varphi \Leftrightarrow \mathcal{M}' \models \varphi$. Since $\mathcal{R}$ is total, it is sufficient to prove that for all $[\mathcal{E}, e]$-free formulas φ of $L(P)$ and all $(w, w') \in \mathcal{R}$: $\mathcal{M}, w \models \varphi \Leftrightarrow \mathcal{M}', w' \models \varphi$. The proof is by induction on φ. In the induction step we are going to need the induction hypothesis for several different choices of $\mathcal{R}, w$ and w', so what we will actually prove by induction on φ is this: For all formulas φ of $L(P)$, if $\mathcal{R}$ is a total bisimulation between information cells $\mathcal{M}$ and $\mathcal{M}'$ on P and $(w, w') \in \mathcal{R}$, then $\mathcal{M}, w \models \varphi \Leftrightarrow \mathcal{M}', w' \models \varphi$.

The base case is when φ is propositional. Then the required follows immediately from [atom], using that $(w, w') \in \mathcal{R}$. For the induction step, we have the following cases of φ: $\neg\psi, \psi \wedge \gamma, X\psi, K\psi, B^{\gamma}\psi$. The first two cases are trivial. So is $X\psi$, as $X\psi \leftrightarrow \psi$ holds on any information cell. For $K\psi$ we reason as follows. Let $\mathcal{R}$ be a total bisimulation between information cells $\mathcal{M}$ and $\mathcal{M}'$ with $(w, w') \in \mathcal{R}$. Using that $\mathcal{R}$ is total and that $\mathcal{M}$ and $\mathcal{M}'$ are both $\sim$-connected we get: $\mathcal{M}, w \models K\psi \Leftrightarrow \forall v \in W\colon \mathcal{M}, v \models \psi \overset{\text{i.h.}}{\Leftrightarrow} \forall v' \in W'\colon \mathcal{M}', v \models \psi \Leftrightarrow \mathcal{M}', w' \models K\psi$.

The case of $B^{\gamma}\psi$ is more involved. Let $\mathcal{M}, \mathcal{M}', \mathcal{R}, w$ and w' be as above. By symmetry, it suffices to prove $\mathcal{M}, w \models B^{\gamma}\psi \Rightarrow \mathcal{M}', w' \models B^{\gamma}\psi$. So assume $\mathcal{M}, w \models B^{\gamma}\psi$, that is, $\mathcal{M}, v \models \psi$ for all $v \in Min_{\leq}\{u \in W \mid \mathcal{M}, u \models \gamma\}$. We need to prove $\mathcal{M}', v' \models \psi$ for all $v' \in Min_{\leq'}\{u' \in W' \mid \mathcal{M}', u' \models \gamma\}$. So let $v' \in Min_{\leq'}\{u' \in W' \mid \mathcal{M}', u' \models \gamma\}$. By definition of $Min_{\leq'}$ this means that:

$$\text{for all } u' \in W', \text{ if } \mathcal{M}', u' \models \gamma \text{ then } v' \leq' u'. \tag{6.1}$$

Choose an $x \in Min_{\leq}\{u \in W \mid u \approx u' \text{ and } (u', v') \in \mathcal{R}\}$. We want to use (6.1) to show that the following holds:

$$\text{for all } u \in W, \text{ if } \mathcal{M}, u \models \gamma \text{ then } x \leq u. \tag{6.2}$$

To prove (6.2), let $u \in W$ with $\mathcal{M}, u \models \gamma$. Choose u' with $(u, u') \in \mathcal{R}$. The induction hypothesis implies $\mathcal{M}', u' \models \gamma$. We now prove that $v' \leq' Min_{\leq'}([u']_{\approx})$. To this end, let $u'' \in [u']_{\approx}$. We need to prove $v' \leq' u''$. Since $u'' \approx u'$, Lemma 6.23 implies that u' and u'' are bisimilar. By induction hypothesis we then get $\mathcal{M}', u'' \models \gamma$.[7] Using (6.1) we now get $v' \leq' u''$, as required. This show $v' \leq' Min_{\leq'}([u']_{\approx})$. We now have $Min_{\leq'}([v']_{\approx}) \leq' v' \leq' Min_{\leq'}([u']_{\approx})$, and hence $v' \preceq u'$. By [back] there is then a v s.t. $(v, v') \in \mathcal{R}$ and $v \preceq u$. By choice of x, $x \leq Min_{\leq}([v]_{\approx})$. Using $v \preceq u$, we now finally get: $x \leq Min_{\leq}([v]_{\approx}) \leq Min_{\leq}([u]_{\approx}) \leq u$. This shows that (6.2) holds.

From (6.2) we can now conclude $x \in Min_{\leq}\{u \in W \mid \mathcal{M}, u \models \gamma\}$ and hence, by original assumption, $\mathcal{M}, x \models \psi$. By choice of x there is an $x' \approx x$ with $(x', v') \in \mathcal{R}$. Since $\mathcal{M}, x \models \psi$ and $x' \approx x$, we can again use Lemma 6.23 and the induction hypothesis to conclude $\mathcal{M}, x' \models \psi$. Since $(x', v') \in \mathcal{R}$, another instance of the induction hypothesis gives us $\mathcal{M}', v' \models \psi$, and we are done. □

Previously we proved that there can only be finitely many non-bisimilar information cells on any finite set P. Since we have now shown that bisimilarity implies modal equivalence, we immediately get the following result, which will be essential to our proof of termination of our planning algorithms.

Corollary 6.25. *Given any finite set P, there are only finitely many modally inequivalent information cells on P.*

6.4.2 Planning Trees

When synthesising plans, we explicitly construct the search space of the problem as a labelled AND-OR tree, a familiar model for planning under uncertainty [Ghallab et al., 2004]. Our AND-OR trees are called *planning trees*.

Definition 6.26 (Planning Tree)**.** A *planning tree* is a finite, labelled AND-OR tree in which each node n is labelled by a plausibility model $\mathcal{M}(n)$, and each edge (n, m) leaving an OR-node is labelled by an event model $\mathcal{E}(n, m)$.

Planning trees for planning problems $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$ are constructed as follows: Let the initial planning tree T_0 consist of just one OR-node $root(T_0)$ with $\mathcal{M}(root(T_0)) = \mathcal{M}_0$ (the root labels the initial state). A planning tree for $\mathcal{P}$ is then any tree that can be constructed from T_0 by repeated applications of the following non-deterministic tree expansion rule.

[7] Note that we here use the induction hypothesis for the autobisimulation on $\mathcal{M}'$ linking u' and u'', not the bisimulation $\mathcal{R}$ between $\mathcal{M}$ and $\mathcal{M}'$.

Definition 6.27 (Tree Expansion Rule)**.** Let T be a planning tree for a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$. The tree expansion rule is defined as follows. Pick an OR-node n in T and an event model $\mathcal{E} \in \mathsf{A}$ applicable in $\mathcal{M}(n)$ with the proviso that $\mathcal{E}$ does not label any existing outgoing edges from n. Then:

1. Add a new AND-node m to T with $\mathcal{M}(m) = \mathcal{M}(n) \otimes \mathcal{E}$, and add an edge (n, m) with $\mathcal{E}(n, m) = \mathcal{E}$.
2. For each information cell $\mathcal{M}'$ in $\mathcal{M}(m)$, add an OR-node m' with $\mathcal{M}(m') = \mathcal{M}'$ and add the edge (m, m').

The tree expansion rule is similar in structure to—and inspired by—the expansion rules used in tableau calculi, e.g. for modal and description logics [Horrocks et al., 2006]. Note that the expansion rule applies only to OR-nodes, and that an applicable event model can only be used once at each node.

Considering single-agent planning a two-player game, a useful analogy for planning trees are game trees. At an OR-node n, the agent gets to pick any applicable action $\mathcal{E}$ it pleases, winning if it ever reaches an information model in which the goal formula holds (see the definition of solved nodes further below). At an AND-node m, the environment responds by picking one of the information cells of $\mathcal{M}(m)$—which of the distinguishable outcomes is realised when performing the action.

Without restrictions on the tree expansion rule, even very simple planning problems might be infinitely expanded (e.g. by repeatedly choosing a no-op action). Finiteness of trees (and therefore termination) is ensured by the following blocking condition.

$\mathcal{B}$ The tree expansion rule may not be applied to an OR-node n for which there exists an ancestor OR-node m with $\mathcal{M}(m) \equiv \mathcal{M}(n)$.[8]

Lemma 6.28 (Termination)**.** *Any planning tree built by repeated application of the tree expansion rule under condition $\mathcal{B}$ is finite.*

Proof. Planning trees built by repeated application of the tree expansion rule are finitely branching: the action library is finite, and every plausibility model has only finitely many information cells (the initial state and all event models in the action library are assumed to be finite, and taking the product update of a

[8] Modal equivalence between information cells can be decided by taking their respective bisimulation contractions and then compare for isomorphism, cf. Section 6.4.1.

finite information cell with a finite event model always produces a finite result). Furthermore, condition $\mathcal{B}$ ensures that no branch has infinite length: there only exists finitely many modally inequivalent information cells over any language $L(P)$ with finite P (Corollary 6.25). König's Lemma now implies finiteness of the planning tree. □

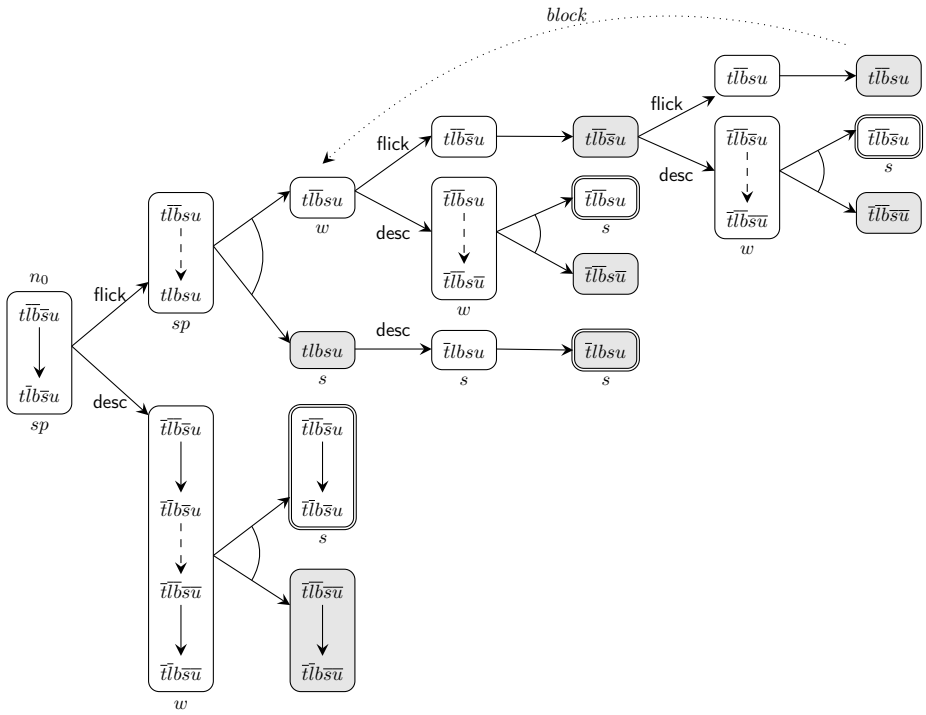


Figure 6.7: A planning tree T for $\mathcal{P}_B$. Each node contains a (visually compacted) plausibility model. Most plausible children of AND-nodes are gray, doubly drawn OR-nodes satisfy the goal formula, and below solved nodes we've indicated their strength.

Example 6.29. Let's consider a planning tree in relation to our basement scenario (cf. Example 6.19). Here the planning problem is $\mathcal{P}_B = (\mathcal{M}_0, \{\mathsf{flick}, \mathsf{desc}\}, \varphi_g)$ with $\mathcal{M}_0$, flick and desc being defined in Figure 6.4 and $\varphi_g = \neg t \wedge u$. We have illustrated the planning tree T in Figure 6.7. The root n_0 is an OR-node (representing the initial state $\mathcal{M}_0$), to which the tree expansion rule of Definition 6.27 has been applied twice, once with action $\mathcal{E} = \mathsf{flick}$ and once with $\mathcal{E} = \mathsf{desc}$.

The result of the two tree expansions on n_0 is two AND-nodes (children of n_0) and four OR-nodes (grandchildren of n_0). We end our exposition of the tree

expansion rule here, and note that the tree has been fully expanded under the blocking condition $\mathcal{B}$, the dotted edge indicating a leaf having a modally equivalent ancestor. Without the blocking condition, this branch could have been expanded ad infinitum.

Let T denote a planning tree containing an AND-node n with a child m. The node m is called a *most plausible child* of n if $\mathcal{M}(m)$ is among the most plausible information cells of $\mathcal{M}(n)$.

Definition 6.30 (Solved Nodes)**.** Let T be any planning tree for a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$. Let α be one of s, w, sp or wp. By recursive definition, a node n in T is called α-*solved* if one of the following holds:

- $\mathcal{M}(n) \models \varphi_g$ (the node satisfies the goal formula).
- n is an OR-node having at least one α-solved child.
- n is an AND-node and:
  - If $\alpha = s$ then all children of n are α-solved.
  - If $\alpha = w$ then at least one child of n is α-solved.
  - If $\alpha = sp$ then all most plausible children of n are α-solved.
  - If $\alpha = wp$ then at least one of the most plausible children of n is α-solved.

Let T denote any planning tree for a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$. Below we show that when an OR-node n of T is α-solved, it is possible to construct an α-solution to the planning problem $(\mathcal{M}(n), \mathsf{A}, \varphi_g)$. In particular, if the root node is α-solved, an α-solution to $\mathcal{P}$ can be constructed. As it is never necessary to expand an α-solved node, nor any of its descendants, we can augment the blocking condition $\mathcal{B}$ in the following way (parameterised by α where α is one of s, w, sp or wp).

$\mathcal{B}_\alpha$ The tree expansion rule may not be applied to an OR-node n if one of the following holds: 1) n is α-solved; 2) n has an α-solved ancestor; 3) n has an ancestor OR-node m with $\mathcal{M}(m) \equiv \mathcal{M}(n)$.

A planning tree that has been built according to $\mathcal{B}_\alpha$ is called an α-*planning tree*. Since $\mathcal{B}_\alpha$ is more strict than $\mathcal{B}$, Lemma 6.28 immediately gives finiteness of α-planning trees—and hence termination of any algorithm building such trees by repeated application of the tree expansion rule. Note that a consequence of $\mathcal{B}_\alpha$ is that in any α-planning tree an α-solved OR-node is either a leaf or has exactly one α-solved child. We make use of this in the following definition.

Definition 6.31 (Plans for Solved Nodes)**.** Let T be any α-planning tree for $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$. For each α-solved node n in T, a plan $\pi(n)$ is defined recursively by:

- if $\mathcal{M}(n) \models \varphi_g$, then $\pi(n) = \mathsf{skip}$.
- if n is an OR-node and m its α-solved child, then $\pi(n) = \mathcal{E}(n, m); \pi(m)$.
- if n is an AND-node and $m_1, \ldots, m_k$ its α-solved children, then
  - If $k = 1$ then $\pi(n) = \pi(m_1)$.
  - If $k > 1$ then for all $i = 1, \ldots, k$ let δ_{m_i} denote a formula true in $\mathcal{M}(m_i)$ but not in any of the $\mathcal{M}(m_j) \not\equiv \mathcal{M}(m_i)$ and let $\pi(n) =$ $\mathsf{if}\ \delta_{m_1}\ \mathsf{then}\ \pi(m_1)\ \mathsf{else}\ \mathsf{if}\ \delta_{m_2}\ \mathsf{then}\ \pi(m_2)\ \mathsf{else}\ \cdots\ \mathsf{if}\ \delta_{m_k}\ \mathsf{then}\ \pi(m_k)$.

Note that the plan $\pi(n)$ of a α-solved node n is only uniquely defined up to the choice of δ-formulas in the if-then-else construct. This ambiguity in the definition of $\pi(n)$ will not cause any troubles in what follows, as it only depends on formulas satisfying the stated property. We need, however, to be sure that such formulas always exist and can be computed. To prove this, assume n is an AND-node and $m_1, \ldots, m_k$ its α-solved children. Choose $i \in \{1, \ldots, k\}$, and let $m_{n_1}, \ldots, m_{n_l}$ denote the subsequence of $m_1, \ldots, m_k$ for which $\mathcal{M}(m_{n_j}) \not\equiv \mathcal{M}(m_i)$. We need to prove the existence of a formula δ_{m_i} such that $\mathcal{M}(m_i) \models \delta_{m_i}$ but $\mathcal{M}(m_{n_j}) \not\models \delta_{m_i}$ for all $j = 1, \ldots, l$. Since $\mathcal{M}(m_{n_j}) \not\equiv \mathcal{M}(m_i)$ for all $j = 1, \ldots, l$, there exists formulas δ_j such that $\mathcal{M}(m_i) \models \delta_j$ but $\mathcal{M}(m_{n_j}) \not\models \delta_j$. We then get that $\delta_1 \wedge \delta_2 \wedge \cdots \wedge \delta_l$ is true in $\mathcal{M}(m_i)$ but none of the $\mathcal{M}(m_{n_j})$. Such formulas can definitely be computed, either by brute force search through all formulas ordered by length or more efficiently and systematically by using characterising formulas as in [Andersen et al., 2012] (however, characterising formulas for the present formalism are considerably more complex than in the purely epistemic framework of the cited paper).

Let n be a node of a planning tree T. We say that n is *solved* if it is α-solved for some α. If n is s-solved then it is also sp-solved, if sp-solved then wp-solved, and if wp-solved then w-solved. This gives a natural ordering $s > sp > wp > w$. Note the relation to Lemma 6.20. We say that a solved node n has *strength* α, if it is α-solved but not β-solved for any $\beta > \alpha$, using the aforementioned ordering.

Example 6.32. Consider again the planning tree T in Figure 6.7 for the planning problem $\mathcal{P}_B = (\mathcal{M}_0, \{\mathsf{flick}, \mathsf{desc}\}, \varphi_g)$ with $\varphi_g = \neg t \wedge u$. Each solved node has been labelled by its strength. The reader is encouraged to check that each node has been labelled correctly according to Definition 6.30. The leafs satisfying the goal formula φ_g have strength s, by definition. The strength of the root

node is *sp*, as its uppermost child has strength *sp*. The reason this child has strength *sp* is that *its* most plausible child has strength *s*.

We see that T is an *sp*-planning tree, as it is possible to achieve T from n_0 by applying tree expansions in an order that respects $\mathcal{B}_{sp}$. However, it is not the smallest *sp*-planning tree for the problem, as e.g. the lower subtree is not required for n_0 to be *sp*-solved. Moreover, T is *not* a *w*-planning tree, as $\mathcal{B}_w$ would have blocked further expansion once either of the three solved leafs were expanded.

In our soundness result below, we show that plans of α-solved roots are always α-solutions to their corresponding planning problems. Applying Definition 6.31 to the *sp*-planning tree T gives an *sp*-solution to the basement planning problem, viz. $\pi(n_0) = \mathsf{flick}; \mathsf{desc}; \mathsf{skip}$. This is the solution we referred to as the *best* in Example 6.1: Assuming all actions result in their most plausible outcomes, the best plan is to flick the switch and then descend. After having executed the first action of the plan, flick, the agent will know whether the bulb is broken or not. This is signified by the two distinct information cells resulting from the flick action, see Figure 6.7. An agent capable of replanning could thus choose to revise her plan and/or goal if the bulb turns out to be broken.

Theorem 6.33 (Soundness)**.** *Let α be one of s, w, sp or wp. Let T be an α-planning tree for a problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$ such that root(T) is α-solved. Then $\pi(root(T))$ is an α-solution to $\mathcal{P}$.*

Proof. We need to prove that $\pi(root(T))$ is an α-solution to $\mathcal{P}$, that is, $\mathcal{M}_0 \models [\pi(root(T))]_\alpha \varphi_g$. Since $\mathcal{M}_0$ is the label of the root, this can be restated as $\mathcal{M}(root(T)) \models [\pi(root(T))]_\alpha \varphi_g$. To prove this fact, we will prove the following stronger claim:

$$\text{For each } \alpha\text{-solved OR-node } n \text{ in } T,\ \mathcal{M}(n) \models [\pi(n)]_\alpha \varphi_g.$$

We prove this by induction on the height of n. The base case is when n is a leaf (height 0). Since n is α-solved, we must have $\mathcal{M}(n) \models \varphi_g$. In this case $\pi(n) = \mathsf{skip}$. From $\mathcal{M}(n) \models \varphi_g$ we can conclude $\mathcal{M}(n) \models [\mathsf{skip}]_\alpha \varphi_g$, that is, $\mathcal{M}(n) \models [\pi(n)]_\alpha \varphi_g$. This covers the base case. For the induction step, let n be an arbitrary α-solved OR-node n of height $h > 0$. Let m denote the α-solved child of n, and $m_1, \ldots, m_l$ denote the children of m. Let $m_{n_1}, \ldots, m_{n_k}$ denote the subsequence of $m_1, \ldots, m_l$ consisting of the α-solved children of m. Then, by Definition 6.31,

- If $k = 1$ then $\pi(n) = \mathcal{E}(n, m); \pi(m_{n_1})$.
- If $k > 1$ then $\pi(n) = \mathcal{E}(n, m); \pi(m)$ where $\pi(m) =$ $\mathsf{if}\ \delta_{m_{n_1}}\ \mathsf{then}\ \pi(m_{n_1})\ \mathsf{else\ if}\ \delta_{m_{n_2}}\ \mathsf{then}\ \pi(m_{n_2})\ \mathsf{else}\ \cdots\ \mathsf{if}\ \delta_{m_{n_k}}\ \mathsf{then}\ \pi(m_{n_k})$.

We here consider only the (more complex) case $k > 1$. Our goal is to prove $\mathcal{M}(n) \models [\pi(n)]_\alpha \varphi_g$, that is, $\mathcal{M}(n) \models [\mathcal{E}(n,m);\pi(m)]_\alpha \varphi_g$. By the induction hypothesis we have $\mathcal{M}(m_{n_i}) \models [\pi(m_{n_i})]_\alpha \varphi_g$ for all $i = 1,\ldots,k$ (the m_{n_i} are of lower height than n).

Claim 1. $\mathcal{M}(m_{n_i}) \models [\pi(m)]_\alpha \varphi_g$ for all $i = 1,\ldots,k$.

Proof of claim. Let i be given. We need to prove

$$\mathcal{M}(m_{n_i}) \models \Big[\mathsf{if}\ \delta_{m_{n_1}}\ \mathsf{then}\ \pi(m_{n_1})\ \mathsf{else}\ \cdots\ \mathsf{if}\ \delta_{m_{n_k}}\ \mathsf{then}\ \pi(m_{n_k})\Big]_\alpha \varphi_g.$$

Note that by using item 5 of Lemma 6.17 it suffices to prove that for all $j = 1,\ldots,k$,

$$\mathcal{M}(m_{n_i}) \models \delta_{m_{n_j}} \text{ implies } \mathcal{M}(m_{n_i}) \models \big[\pi(m_{n_j})\big]_\alpha \varphi_g. \tag{6.3}$$

Let $j \in \{1,\ldots,k\}$ be chosen arbitrarily. Assume first $j = i$. By induction hypothesis we have $\mathcal{M}(m_{n_j}) \models \big[\pi(m_{n_j})\big]_\alpha \varphi_g$, and hence $\mathcal{M}(m_{n_i}) \models \big[\pi(m_{n_j})\big]_\alpha \varphi_g$. From this (6.3) immediately follows. Assume now $j \neq i$. By the construction of the δ-formulas, either $\mathcal{M}(m_{n_j}) \equiv \mathcal{M}(m_{n_i})$ or $\mathcal{M}(m_{n_i}) \not\models \delta_{m_{n_j}}$. In the latter case, (6.3) holds trivially. In case of $\mathcal{M}(m_{n_j}) \equiv \mathcal{M}(m_{n_i})$ we immediately get $\mathcal{M}(m_{n_i}) \models \big[\pi(m_{n_j})\big]_\alpha \varphi_g$, since by induction hypothesis we have $\mathcal{M}(m_{n_j}) \models \big[\pi(m_{n_j})\big]_\alpha \varphi_g$. This concludes the proof of the claim.

Note that by definition of the tree expansion rule (Definition 6.27), $\mathcal{M}(m_1),\ldots,\mathcal{M}(m_l)$ are the information cells in $\mathcal{M}(m)$.

Claim 2. The following holds:

- If $\alpha = s$ (w), then for every (some) information cell $\mathcal{M}'$ in $\mathcal{M}(m)$: $\mathcal{M}' \models [\pi(m)]_\alpha \varphi_g$.
- If $\alpha = sp$ (wp), then for every (some) most plausible information cell $\mathcal{M}'$ in $\mathcal{M}(m)$: $\mathcal{M}' \models [\pi(m)]_\alpha \varphi_g$.

Proof of claim. We only consider the most complex cases, $\alpha = sp$ and $\alpha = wp$. First consider $\alpha = sp$. Let $\mathcal{M}'$ be a most plausible information cell in $\mathcal{M}(m)$. We need to prove $\mathcal{M}' \models [\pi(m)]_\alpha \varphi_g$. Since, as noted above, $\mathcal{M}(m_1),\ldots,\mathcal{M}(m_l)$ are the information cells in $\mathcal{M}(m)$, we must have $\mathcal{M}' = \mathcal{M}(m_i)$ for some $i \in \{1,\ldots,l\}$. Furthermore, as $\mathcal{M}'$ is among the most plausible information cells in $\mathcal{M}(m)$, m_i must by definition be a most plausible child of m. Definition 6.30 then gives us that m_i is α-solved. Thus $m_i = m_{n_j}$ for some $j \in \{1,\ldots,k\}$. By Claim 1 we have $\mathcal{M}(m_{n_j}) \models [\pi(m)]_\alpha \varphi_g$, and since $\mathcal{M}' = \mathcal{M}(m_i) = \mathcal{M}(m_{n_j})$ this gives the desired conclusion. Now consider the case $\alpha = wp$. Definition 6.30

gives us that at least one of the most plausible children of m are α-solved. By definition, this must be one of the m_{n_i}, $i \in \{1, \ldots, k\}$. Claim 1 gives $\mathcal{M}(m_{n_i}) \models [\pi(m)]_\alpha \varphi_g$. Since m_{n_i} is a most plausible child of m, we must have that $\mathcal{M}(m_{n_i})$ is among the most plausible information cells in $\mathcal{M}(m)$. Hence we have proven that $[\pi(m)]_\alpha \varphi_g$ holds in a most plausible information cell of $\mathcal{M}(m)$.

By definition of the tree expansion rule (Definition 6.27), $\mathcal{M}(m) = \mathcal{M}(n) \otimes \mathcal{E}(n, m)$. Thus we can replace $\mathcal{M}(m)$ by $\mathcal{M}(n) \otimes \mathcal{E}(n, m)$ in Claim 2 above. Using items 1–4 of Lemma 6.17, we immediately get from Claim 2 that independently of α the following holds: $\mathcal{M}(n) \models [\mathcal{E}(n, m)]_\alpha [\pi(m)]_\alpha \varphi_g$ (the condition $\mathcal{M}(n) \models \langle \mathcal{E}(n, m) \rangle \top$ holds trivially by the tree expansion rule). From this we can then finally conclude $\mathcal{M}(n) \models [\mathcal{E}(n, m); \pi(m)]_\alpha \varphi_g$, as required. □

Theorem 6.34 (Completeness)**.** *Let α be one of s, w, sp or wp. If there is an α-solution to the planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$, then an α-planning tree T for $\mathcal{P}$ can be constructed, such that root(T) is α-solved.*

Proof. First note that we have $[\mathsf{skip}; \pi]_\alpha \varphi_g = [\mathsf{skip}]_\alpha ([\pi]_\alpha \varphi_g) = [\pi]_\alpha \varphi_g$. Thus, we can without loss of generality assume that no plan contains a subexpression of the form $\mathsf{skip}; \pi$. The length of a plan π, denoted $|\pi|$, is defined recursively by: $|\mathsf{skip}| = 1$; $|\mathcal{E}| = 1$; $|\mathsf{if}\ \varphi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2| = |\pi_1| + |\pi_2|$; $|\pi_1; \pi_2| = |\pi_1| + |\pi_2|$.

Claim 1. Let π be an α-solution to $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$ with $|\pi| \geq 2$. Then there exists an α-solution of the form $\mathcal{E}; \pi'$ with $|\mathcal{E}; \pi'| \leq |\pi|$.

Proof of claim. Proof by induction on $|\pi|$. The base case is $|\pi| = 2$. We have two cases, $\pi = \mathsf{if}\ \varphi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ and $\pi = \pi_1; \pi_2$, both with $|\pi_1| = |\pi_2| = 1$. If π is the latter, it already has desired the form. If $\pi = \mathsf{if}\ \varphi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ then, by assumption on π, $\mathcal{M}_0 \models [\mathsf{if}\ \varphi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2]_\alpha \varphi_g$. Item 5 of Lemma 6.17 now gives that $\mathcal{M}_0 \models \varphi$ implies $\mathcal{M}_0 \models [\pi_1]_\alpha \varphi_g$ and $\mathcal{M}_0 \not\models \varphi$ implies $\mathcal{M}_0 \models [\pi_2]_\alpha \varphi_g$. Thus we must either have $\mathcal{M}_0 \models [\pi_1]_\alpha \varphi_g$ or $\mathcal{M}_0 \models [\pi_2]_\alpha \varphi_g$, that is, either π_1 or π_2 is an α-solution to $\mathcal{P}$. Thus either $\pi_1; \mathsf{skip}$ or $\pi_2; \mathsf{skip}$ is an α-solution to $\mathcal{P}$, and both of these have length $|\pi|$. This completes the base case. For the induction step, consider a plan π of length $l > 2$ which is an α-solution to $\mathcal{P}$. We again have two cases to consider, $\pi = \mathsf{if}\ \varphi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ and $\pi = \pi_1; \pi_2$. If $\pi = \pi_1; \pi_2$ is an α-solution to $\mathcal{P}$, then π_1 is an α-solution to the planning problem $\mathcal{P}' = (\mathcal{M}_0, \mathsf{A}, [\pi_2]_\alpha \varphi_g)$, as $\mathcal{M}_0 \models [\pi_1; \pi_2]_\alpha \varphi_g \Leftrightarrow \mathcal{M}_0 \models [\pi_1]_\alpha [\pi_2]_\alpha \varphi_g$. Clearly $|\pi_1| < l$, so the induction hypothesis gives that there is an α-solution $(\mathcal{E}; \pi_1')$ to $\mathcal{P}'$, with $|\mathcal{E}; \pi_1'| \leq |\pi_1|$. Then, $\mathcal{E}; \pi_1'; \pi_2$ is an α-solution to $\mathcal{P}$ and we have $|\mathcal{E}; \pi_1'; \pi_2| = |\mathcal{E}; \pi_1'| + |\pi_2| \leq |\pi_1| + |\pi_2| = |\pi|$. If $\pi = \mathsf{if}\ \varphi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ is an α-solution to $\mathcal{P}$, then we can as above conclude that either π_1 or π_2 is an α-solution to $\mathcal{P}$. With both $|\pi_1| < l$ and $|\pi_2| < l$, the induction hypothesis gives

the existence an α-solution $\mathcal{E};\pi'$, with $|\mathcal{E};\pi'| \leq |\pi|$. This completes the proof of the claim.

We now prove the theorem by induction on $|\pi|$, where π is an α-solution to $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$. We need to prove that there exists an α-planning tree for $\mathcal{P}$ in which the root is α-solved. Let T_0 denote the planning tree for $\mathcal{P}$ only consisting of its root node with label $\mathcal{M}_0$. The base case is when $|\pi| = 1$. Here, we have two cases, $\pi = \mathsf{skip}$ and $\pi = \mathcal{E}$. In the first case, the planning tree T_0 already has its root α-solved, since $\mathcal{M}_0 \models [\mathsf{skip}]_\alpha \varphi_g \Leftrightarrow \mathcal{M}_0 \models \varphi_g$. In the second case, $\pi = \mathcal{E}$, we have $\mathcal{M}_0 \models [\mathcal{E}]_\alpha \varphi_g$ as $\pi = \mathcal{E}$ is an α-solution to $\mathcal{P}$. By definition, this means that $\mathcal{E}$ is applicable in $\mathcal{M}_0$, and we can apply the tree expansion rule to T_0, which will produce:

1. A child m of the root node with $\mathcal{M}(m) = \mathcal{M}_0 \otimes \mathcal{E}$.
2. Children $m_1, \ldots, m_l$ of m, where $\mathcal{M}(m_1), \ldots, \mathcal{M}(m_l)$ are the information cells of $\mathcal{M}(m)$.

Call the expanded tree T_1. Since $\mathcal{M}_0 \models [\mathcal{E}]_\alpha \varphi_g$, Lemma 6.17 implies that for every/some/every most plausible/some most plausible information cell $\mathcal{M}'$ in $\mathcal{M}_0 \otimes \mathcal{E}$, $\mathcal{M}' \models \varphi_g$ (where $\alpha = s/w/sp/wp$). Since $\mathcal{M}(m_1), \ldots, \mathcal{M}(m_l)$ are the information cells of $\mathcal{M}_0 \otimes \mathcal{E}$, we can conclude that every/some/every most plausible/some most plausible child of m is α-solved. Hence also m and thus n are α-solved. The base is hereby completed.

For the induction step, let π be an α-solution to $\mathcal{P}$ with length $l > 1$. Let T_0 denote the planning tree for $\mathcal{P}$ consisting only of its root node with label $\mathcal{M}_0$. By Claim 1, there exists an α-solution to $\mathcal{P}$ of the form $\mathcal{E};\pi'$ with $|\mathcal{E};\pi'| \leq |\pi|$. As $\mathcal{M}_0 \models [\mathcal{E};\pi']_\alpha \varphi_g \Leftrightarrow \mathcal{M}_0 \models [\mathcal{E}]_\alpha [\pi']_\alpha \varphi_g$, $\mathcal{E}$ is applicable in $\mathcal{M}_0$. Thus, as in the base case, we can apply the tree expansion rule to T_0 which will produce nodes as in 1 and 2 above. Call the expanded tree T_1. Since $\mathcal{M}_0 \models [\mathcal{E}]_\alpha [\pi']_\alpha \varphi_g$, items 1–4 of Lemma 6.17 implies that for every/some/every most plausible/some most plausible information cell in $\mathcal{M}_0 \otimes \mathcal{E}$, $[\pi']_\alpha \varphi_g$ holds. Hence, for every/some/every most plausible/some most plausible child m_i of m, $\mathcal{M}(m_i) \models [\pi']_\alpha \varphi_g$. Let $m_{n_1}, \ldots, m_{n_k}$ denote the subsequence of $m_1, \ldots, m_l$ consisting of the children of m for which $\mathcal{M}(m_{n_i}) \models [\pi']_\alpha \varphi_g$. Then, by definition, π' is an α-solution to each of the planning problem $\mathcal{P}_i = (\mathcal{M}(m_{n_i}), \mathsf{A}, \varphi_g)$, $i = 1, \ldots, k$. As $|\pi'| < |\mathcal{E};\pi'| \leq l$, the induction hypothesis gives that α-planning trees T_i' with α-solved roots can be constructed for each $\mathcal{P}_i$. Let T_2 denote T_1 expanded by adding each planning tree T_i' as the subtree rooted at $\mathcal{M}_{n_i}$. Then each of the nodes m_{n_i} are α-solved in T, and in turn both m and $root(T_2)$ are α-solved. The final thing we need to check is that T_2 has been correctly constructed according to the tree expansion rule, more precisely, that condition $\mathcal{B}_\alpha$ has not been violated.

Since each T_i' has in itself been correctly constructed in accordance with $\mathcal{B}_\alpha$, the condition can only have been violated if for one of the non-leaf OR-nodes m' in one of the T_i's, $\mathcal{M}(m') \equiv \mathcal{M}(root(T_2))$. We can then replace the entire planning tree T_2 by a (node-wise modally equivalent) copy of the subtree rooted at m', and we would again have an α-planning tree with an α-solved root. □

6.4.3 Planning Algorithm

In the following, let $\mathcal{P}$ denote any planning problem, and α be one of s, w, sp or wp. With all the previous in place, we now have an algorithm for synthesising an α-solution to $\mathcal{P}$, given as follows.

PLAN$(\alpha, \mathcal{P})$

1 Let T be the α-planning tree only consisting of $root(T)$ labelled by the initial state of $\mathcal{P}$.

2 Repeatedly apply the tree expansion rule of $\mathcal{P}$ to T until no more rules apply satisfying condition $\mathcal{B}_\alpha$.

3 If $root(T)$ is α-solved, return $\pi(root(T))$, otherwise return FAIL.

Theorem 6.35. PLAN$(\alpha, \mathcal{P})$ *is a terminating, sound and complete algorithm for producing α-solutions to planning problems $\mathcal{P}$. Soundness means that if* PLAN$(\alpha, \mathcal{P})$ *returns a plan, it is an α-solution to $\mathcal{P}$. Completeness means that if $\mathcal{P}$ has an α-solution,* PLAN$(\alpha, \mathcal{P})$ *will return one.*

Proof. Termination comes from Lemma 6.28 (with $\mathcal{B}$ replaced by the stronger condition $\mathcal{B}_\alpha$), soundness from Theorem 6.33 and completeness from Theorem 6.34 (given any two $\mathcal{B}_\alpha$-saturated α-planning trees T_1 and T_2 for the same planning problem, the root node of T_1 is α-solved iff the root node of T_2 is). □

With PLAN$(\alpha, \mathcal{P})$ we have given an algorithm for solving α-parametrised planning problems. The α parameter determines the strength of the synthesised plan π, cf. Lemma 6.20. Whereas the cases of weak ($\alpha = w$) and strong ($\alpha = s$) plans have been the subject of much research, the generation of weak plausibility ($\alpha = wp$) and strong plausibility ($\alpha = sp$) plans based on pre-encoded beliefs is a novelty of this paper. Plans taking plausibility into consideration have several advantages. Conceptually, the basement scenario as formalised by $\mathcal{P}_B$ (cf. Example 6.19) allowed for several weak solutions (with the shortest one being hazardous to the agent) and *no* strong solutions. In this case, the synthesised strong plausibility solution corresponds to the course of action a rational agent (mindful of her beliefs) should take. There are also computational advantages.

An invocation of $\textsc{Plan}(sp, \mathcal{P})$ will expand at most as many nodes as an invocation of $\textsc{Plan}(s, \mathcal{P})$ before returning a result (assuming the same order of tree expansions). As plausibility plans only consider the most plausible information cells, we can prune non-minimal information cells during plan search.

We also envision using this technique in the context of an agent framework where planning, acting and execution monitoring are interleaved.[9] Let us consider the case of strong plausibility planning ($\alpha = sp$). From some initial situation an sp-plan is synthesised which the agent starts executing. If reaching a situation that is not covered by the plan, she restarts the process from this point; i.e. she *replans*. Note that the information cell to replan from is present in the tree as a sibling of the most plausible information cell(s) expected from executing the last action. Such replanning mechanisms allow for the *repetition* of actions necessary in some planning problems with cyclic solutions.

We return one last time to the basement problem and consider a modified replace action such that the replacement light bulb might, though it is unlikely, be broken. This means that there is no strong solution. Executing the sp-solution flick; desc, she would replan after flick if that action didn't have the effect of turning on the light. A strong plausibility solution from this point would then be flick; replace; flick; desc.

6.5 Related and Future Work

In this paper we have presented α-solutions to planning problems incorporating ontic, epistemic and doxastic notions. The cases of $\alpha = sp/sw$ are, insofar as we are aware, novel concepts not found elsewhere in the literature. Our previous paper [Andersen et al., 2012] concerns the cases $\alpha = s/w$, so that framework deals only with *epistemic* planning problems without a doxastic component. Whereas we characterise solutions as formulas, [Bolander and Andersen, 2011] takes a semantic approach to strong solutions for epistemic planning problems. In their work plans are sequences of actions, requiring conditional choice of actions at different states to be encoded in the action structure itself. By using the $\mathcal{L}(P, \mathsf{A})$ we represent this choice explicitly.

The meaningful plans of [de Lima, 2007, chap. 2] are reminiscent of the work in this paper. Therein, plan verification is cast as validity of an EDL-consequence in a given system description. Like us, they consider single-agent scenarios, conditional plans, applicability and incomplete knowledge in the initial state.

[9] Covering even more mechanisms of agency is *situated planning* [Ghallab et al., 2004].

Unlike us, they consider only deterministic epistemic actions (without plausibility). In the multi-agent treatment [de Lima, 2007, chap. 4], action laws are translated to a fragment of DEL with only public announcements and public assignments, making actions singleton event models. This means foregoing nondeterminism and therefore sensing actions.

Epistemic planning problems in [Löwe et al., 2011] are solved by producing a sequence of pointed epistemic event models where an external variant of applicability (called *possible at*) is used. Using such a formulation means outcomes of actions are fully determined, making conditional plans and weak solutions superfluous. As noted by the authors, and unlike our framework, their approach does not consider factual change. We stress that [Bolander and Andersen, 2011, Löwe et al., 2011, de Lima, 2007] all consider the multi-agent setting which we have not treated here.

In our work so far, we haven't treated the problem of where domain formulations come from, assuming just that they are given. Standardised description languages are vital if modal logic-based planning is to gain wide acceptance in the planning community. Recent work worth noting in this area includes [Baral et al., 2012], which presents a specification language for the multi-agent belief case.

As suggested by our construction of planning trees, there are several connections between our approach for $\alpha = s$ and two-player imperfect information games. First, product updates imply perfect recall [van Benthem, 2001]. Second, when the game is at a node belonging to an information set, the agent knows a proposition only if it holds throughout the information set. Finally, the strong solutions we synthesise are very similar to mixed strategies. A strong solution caters to any information cell (contingency) it may bring about, by selecting exactly one sub-plan for each [Aumann and Hart, 1992].

Our work relates to [Ghallab et al., 2004], where the notions of strong and weak solutions are found, but without plausibilites. Their belief states are sets of states which may be partioned by observation variables. The framework in [Rintanen, 2004] describes strong conditional planning (prompted by nondeterministic actions) with partial observability modelled using a fixed set of observable state variables. Our partition of plausibility models into information cells follows straight from the definition of product update. A clear advantage in our approach is that *actions* readily encode both nondetermism and partial observability. [Jensen, 2013a] shows that the *strong plan existence problem* for the framework in [Andersen et al., 2012] is 2-EXP-complete.[10] In our formulation, $\textsc{Plan}(s, \mathcal{P})$ answers the same question for $\mathcal{P}$ (it gives a strong solution if

[10] See Chapter 3

one exists), though with a richer modal language.

We would like to do plan verification and synthesis in the multi-agent setting. We believe that generalising the notions introduced in this paper to multi-pointed plausibility and event models are key. Plan synthesis in the multi-agent setting is undecidable [Bolander and Andersen, 2011], but considering restricted classes of actions as is done in [Löwe et al., 2011] seems a viable route for achieving decidable multi-agent planning. Other ideas for future work include replanning algorithms and learning algorithms where plausibilities of actions can be updated when these turn out to have different outcomes than expected.

Acknowledgements

For valuable comments at various stages of the work presented in this article, we would like to extend our gratitude to the following persons: Patrick Allo, Alexandru Baltag, Johan van Benthem, Hans van Ditmarsch, Jens Ulrik Hansen, Sonja Smets and the anonymous reviewers.

Chapter 7

Conclusion and Discussion

We started this thesis by lifting the foundation of automated planning to single-agent dynamic epistemic logic. Then we showed that this formalism is able express planning with or without branching actions as well as with full observability or partial observability. Our complexity results prove that our formalism is no worse in terms of complexity than that used elsewhere in the field of automated planning. Next we took at look at single-agent epistemic-plausibility models, which are structures that allow us to reason about doxastic notions. We presented a notion of bisimulation that corresponds to modal equivalence (in the typical sense) for the doxastic language with both a conditional belief modality as well as a degrees of belief modality. Surprisingly, it turns out that these two doxastic notions are incomparable over an infinite set of propositional symbols. Lastly, we formalized solution concepts that only require plans to cater for the expected outcomes. For an extensive account of our results we refer the reader to Section 1.6.

What we find to be an important result methodologically is that by lifting the foundation of automated planning to dynamic epistemic logic, we were able to directly utilize results from epistemic model theory. This allowed us to to limit the size of information cells, as well as give an upper bound on the number information cells that we must explore in order to find a solution. In itself this result only allows us to show conditional epistemic planning is an appropriate alternative formalism for automated planning. More interesting is that by taking the first steps in this direction, we were able to develop a new formalism for planning, this time by relying on a dynamic version of doxastic logic. This demonstrates a general method for extending planning, based on research conducted within the field of modal logic.

We must concede to not having handled the highly interesting multi-agent setting, which is arguably where planning founded on dynamic epistemic-doxastic logics shines. There are a plethora of problems that must be addressed to facilitate multi-agent planning. We already mentioned the issue of undecidability in Section 1.3, with the recent [Yu et al., 2013] giving some hope of finding more decidable fragments. Another is finding an adaquate description language that can play the same role as that of PDDL in automated planning (see Section 1.1). Namely, a language that would allow for benchmarking and comparing different approaches to multi-agent planning; one proposal for this is [Baral et al., 2012]. In any case, it seems likely that significant compromises must be made in terms of expressive power, in order for multi-agent planning to ever become practical. What we can at least take away from this fact is that it fully supports our initial view that no single method is universally best for all tasks, and hope that even more research will be conducted in the crossroads between planning and modal logics.

Appendix A

Proofs

A.1 Proofs from Section 3.2

For Lemmas A.1 through A.5 we consider some $k \in \mathbb{N}$, and let T be a planning tree for $\mathcal{P}^k = (\mathcal{M}_0, \mathsf{A}, p_{k+2})$ on $P = \{p_1, \ldots, p_{k+2}\}$ (cf. Definition 3.25).

Lemma A.1. *For any* OR*-node n of T, the label $\mathcal{M}(n)$ denotes a singleton epistemic model satisfying exactly one propositional symbol in P.*

Proof. By inspection of the initial state and action library. We have that $\mathcal{M}(root(T)) = \mathcal{M}_0$ is a singleton epistemic model satisfying exactly p_1. Therefore we have for $1 \leq i \leq k$ that when grow_i is applicable the tree expansion rule results in two OR-nodes being added to T, where one satisfies exactly p_{i+1} and the other satisfies exactly p_{i+2}. Similarly when stop_{k+1} is applicable one OR-node results satisfying exactly p_{k+2}. □

Lemma A.2. *For any* OR*-node n of T, if $\mathcal{M}(n) \models p_{k+2}$ then no actions in A are applicable in $\mathcal{M}(n)$, and otherwise exactly one action in A is applicable in $\mathcal{M}(n)$.*

Proof. By inspection of the action library we have that for $1 \leq i \leq k$, grow_i is applicable iff $\mathcal{M}(n) \models p_i$, and stop_{k+1} is applicable iff $\mathcal{M}(n) \models p_{k+1}$, which covers all actions in A. From Lemma A.1 we have that $\mathcal{M}(n)$ satisfies exactly one propositional symbol, hence the result follows. □

Lemma A.3. *For any* OR*-node n of T, if n is solved then every descendant of n is solved.*

Proof. By induction on the height of n. The base case is when the height of n is 0 (n is a leaf), meaning that n has no descendants so the result holds trivially. For the induction step assume that n has height $h+1$. As n is not a leaf there has to be an action in A which has been used for the expansion of n. By Lemma A.2 only one such $\mathcal{E}$ exists in the action library, and consequently n has exactly one child m of height h. By assumption we have that n is solved, and so as n is an OR-node we have by Definition 2.21 that its single child m must be solved. As a result of m being a solved AND-node it follows that every child n' of m is solved. As each n' is an OR-node whose height is $< h$, we can apply the induction hypothesis to conclude that every descendant of n' is solved. Consequently we have that every descendant of n is solved, showing the induction step and thus completing the proof. □

Lemma A.4. *For any two* OR*-nodes n and n' of T, if n is an ancestor of n', then $\mathcal{M}(n)$ and $\mathcal{M}(n')$ are not bisimilar.*

Proof. Consider every OR-node on the unique path from n to n'. Using Lemma A.1 we have that each such node is labelled with a singleton epistemic model satisfying exactly some p_i. We can therefore identify this path with a sequence of numbers, corresponding to the single propositional symbol satisfied at each OR-node. By inspection of the action library (as in the proof of Lemma A.1), it follows that this is a strictly increasing sequence (by either 1 or 2). Therefore n and n' do not satisfy the same atoms, hence they are not bisimilar. □

Lemma A.5 (Extended proof of Lemma 3.27)**.** *$\mathcal{B}_2$ does not prevent the tree expansion rule from being applied to any* OR*-node n of a planning tree T for $\mathcal{P}^k$*

Proof. For $\mathcal{B}_2$ to prevent n from being expanded, there has to be an action $\mathcal{E}$ in A s.t. $\mathcal{E}$ is applicable in $\mathcal{M}(n)$ and $\mathcal{E}$ has not been used for the expansion of n. Furthermore at least one subcondition of $\mathcal{B}_2$ must be true. We will show this cannot be the case, by assuming in turn each subcondition of $\mathcal{B}_2$ and deriving a contradiction. 1) Assume that n is solved. Since $\mathcal{E}$ is an applicable action it follows from Lemma A.2 that $\mathcal{M}(n) \not\models p_{k+2}$. Therefore as n is a solved OR-node not satisfying the goal formula, it must have a child that is solved. But from Lemma A.2 we have that $\mathcal{E}$ is the only action applicable in $\mathcal{M}(n)$, hence as n has not been expanded by $\mathcal{E}$ it has no children, contradicting that n is solved. 2) Assume that n has a solved ancestor. By Lemma A.3 this implies that n is solved, hence we can derive a contradiction as in 1). 3) Assume that n has an ancestor node n' s.t. $\mathcal{M}(n)$ and $\mathcal{M}(n')$ are bisimilar. By Lemma A.4 such an ancestor cannot exist, hence we have an immediate contradiction. □

A.2 Proofs from Section 3.3

Lemma A.6 (Extended proof of Lemma 3.36)**.** *Let a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$ be given. There exists a $\mathcal{B}_2$-saturated planning tree T for $\mathcal{P}$ whose root is solved iff the saturated planning graph G for $\mathcal{P}$ is solved.*

Proof. We show that if T is $\mathcal{B}_2$-saturated planning tree for $\mathcal{P}$ such that $root(T)$ is solved, then the saturated planning graph $G = (N, M, E, n_0)$ for $\mathcal{P}$ is solved. We do so by constructing from T a graph G', which can be mapped to a solved subgraph of G. The mapping is simple but necessary, as G' is a solved subgraph of G up to bisimilarity. In constructing G' we first prune T so that it contains only solved nodes, and recall that in this pruned tree the outdegree of each OR-node is 1, except nodes satisfying φ_g whose outdegree is 0. We then iteratively construct $G' = (N', M', E', n'_0)$, with $n'_0 = root(T)$ being added at the very end. Adding a node of T to G' means we add it to the corresponding bipartition, that is, OR-nodes of T are added to N' and AND-nodes of T are added to M'. Further, we assign nodes the same label as given in T.

Let h_r be the height $root(T)$ (i.e. the height of T), and N_i, resp. M_i, be the OR-nodes, resp. AND-nodes, in T of height i. Starting from $i = 0$ and in increments of 1, repeatedly apply the operation AddLayer(i) until $i = h_r/2$ (note that the height of an OR-node is an even number and that the height of an AND-node is an odd number).

AddLayer(i)

```
1  for each n ∈ N_2i                                  ▷ Only OR-nodes have this height
2      if G' contains no OR-node totally bisimilar to n then
3          Add n to G'
4          if n has a child m_c in T then
5              m ← AND-node in G' totally bisimilar to m_c
6              Add (n, m_c) with label E(n, m_c) to G'
7  for each m ∈ M_2i+1                                ▷ Only AND-nodes have this height
8      Add m to G'
9      for each child n_c of m
10         n ← OR-node in G' totally bisimilar to n_c
11         Add (m, n) to G'
```

The final call AddLayer($h_r/2$) adds exactly $n'_0 = root(T)$ to G', since $\mathcal{B}_2$-saturation implies that no OR-node in T is bisimilar to $root(T)$ and because $M_{h_r+1} = \emptyset$. Because we do this in a bottom-up fashion based on the height of a node in T, we can always find the totally bisimilar nodes in line 4 and 10 (or equivalently, G' always contains nodes which are totally bisimilar to the

children of nodes chosen in line 1 and 7). Upon completion of these operations we remove every node in G' not reachable from n'_0.

We cannot assume that the labels of OR-nodes in T are necessarily found in G, meaning that G' is not generally a solved subgraph of G for n_0. However, we can use G' to show that a solved subgraph of G exists. To this end, we first show that G' satisfies $\mathcal{C}_1$ and $\mathcal{C}_2$. The criteria $\mathcal{C}_1$ is immediate as every node not reachable from n_0 is removed after the ADDLAYER operations have been carried out, and so we turn to $\mathcal{C}_2$.

Claim 1. G' is acyclic.

Proof of claim. Let G'_j denote G' after j ADDLAYER operations have been carried out. We show that G'_j is acyclic for any $j \in \mathbb{N}$, and proceed by induction on j. Since G'_0 is an empty graph we can immediately proceed to the induction step. From the induction hypothesis it follows that G'_j is acyclic, and so we must show that G'_{j+1} is acyclic. For G'_{j+1} to contain a cycle C, it follows from acyclicity of G'_j that C must visit a node *not* in G'_j. Since the nodes added in line 3 and 8 of ADDLAYER are not reachable from any node in G'_j, we have the stronger requirement that C *only* visit nodes added by ADDLAYER$(j+1)$. The head of each edge added in line 6 is a node in G'_j, meaning that C only visits nodes added in line 8. These nodes have indegree 0, and consequently C cannot exists, hence G'_{j+1} is acyclic. We can now conclude that $G'_{h_r/2} = G'$ is acyclic.

The following claim shows that we can match any node and edge in G_b to a node and edge in G modulo bisimilarity.

Claim 2. Let v'_1 be a node in G'. There is a v_1 in G s.t. $\mathcal{M}(v'_1) \leftrightarrows \mathcal{M}(v_1)$. Further, if v'_2 is a direct successor of v'_1, then v_1 has a direct successor s.t. $\mathcal{M}(v'_2) \leftrightarrows \mathcal{M}(v_2)$.

Proof of claim. Proof is by induction on the length l of the shortest path from n'_0 to v'_1. For the base case we have $l = 0$ and so $v'_1 = n'_0$. We have that $\mathcal{M}(n'_0) = \mathcal{M}(root(T)) = \mathcal{M}(n_0)$, immediately showing that n_0 is a node of G that is bisimilar to n'_0. As n'_0 is an OR-node, then if it has a direct successor v'_2 the label $\mathcal{E}(n'_0, v'_2)$ is an action that is applicable in $\mathcal{M}(n'_0)$. By Corollary 3.7 and $\mathcal{M}(n'_0) \leftrightarrows \mathcal{M}(n_0)$ it follows that $\mathcal{E}(n'_0, v'_2)$ is applicable in $\mathcal{M}(n_0)$. As G it saturated this means n_0 has a direct successor v_2 s.t. $\mathcal{M}(v_2) = \mathcal{M}(n_0) \otimes \mathcal{E}(n'_0, v'_2)$, and so from Lemma 3.8 it follows that $\mathcal{M}(v'_2) \leftrightarrows \mathcal{M}(v_2)$.

For the induction step, assume that the shortest path from n'_0 to v'_1 is of length $l+1$. We therefore have a direct predecessor v'_p of v'_1, and the length of the shortest path from n'_0 to v'_p is l. Applying the induction hypothesis (v'_1 is a direct successor of v'_p) this means G contains nodes v_p, v_1 s.t. v_1 is a direct successor

of v_p, $\mathcal{M}(v'_p) \leftrightarrows \mathcal{M}(v_p)$ and $\mathcal{M}(v'_1) \leftrightarrows \mathcal{M}(v_1)$. It therefore remains to be shown that if v'_2 is a direct successor of v'_1, then there is a direct successor v_2 of v_1 s.t. $\mathcal{M}(v'_2) \leftrightarrows \mathcal{M}(v_2)$. Assuming such a v'_2 exists, this gives rise to two cases. If v'_1 is an OR-node, it follows that $\mathcal{E}(v'_1, v'_2)$ is applicable in v'_1. As in the base this means there is a direct successor v_2 of v_1 s.t. $\mathcal{M}(v_2) = \mathcal{M}(v_1) \otimes \mathcal{E}(v'_1, v'_2)$, and so we have $\mathcal{M}(v'_2) \leftrightarrows \mathcal{M}(v_2)$ as required. If v'_1 is an AND-node it follows that $\mathcal{M}(v'_2)$ is an information cell of $\mathcal{M}(v'_1)$. Total bisimilarity between $\mathcal{M}(v'_1)$ and $\mathcal{M}(v_1)$ means there exists an information cell of $\mathcal{M}(v_1)$ that is bisimilar to $\mathcal{M}(v'_2)$, hence there is a direct successor v_2 of v_1 s.t. $\mathcal{M}(v'_2) \leftrightarrows \mathcal{M}(v_2)$, thereby completing the induction step.

Completing the proof we show that G' can be mapped to a solved subgraph of G for n_0 with $\mathcal{M}(n_0) = \mathcal{M}(root(T))$. We map each node in G' to the node in G' for which it is bisimilar, which is possible by Claim 2. Furthermore, each edge of G' is mapped to an edge in G in the obvious manner, which is possible as both planning trees and planning graphs are expanded based on the product update operation. Let H be the subgraph of G induced by this mapping. As G' satisfies $\mathcal{C}_1$ and $\mathcal{C}_2$ so does H. To see that H satisfies $\mathcal{C}_3$, recall that T was pruned to contain only solved nodes. Therefore any OR-node added to G' has outdegree 0 if it satisfies φ_g and otherwise it has outdegree 1 due to line 4 and 5 of AddLayer, and furthermore no two OR-nodes of G' are totally bisimilar due to line 2. We have that each child of an AND-node in T corresponds to an information cell. Therefore line 9 and 10 of AddLayer means that the direct successors of an AND-node m' in G' are exactly the information cells of $\mathcal{M}(m')$, and so H satisfies $\mathcal{C}_4$. Consequently H is a solved subgraph of G for n_0 as required. □

Recall that G_i denotes the graph produced i iterations of the loop in line 3 of SolvedSubgraph, and that as SolvedSubgraph never removes nodes, G_i is a subgraph of G_{i+1}.

Lemma A.7. *$G_i = (N_i, M_i, E_i)$ is a subgraph of G that satisfies $\mathcal{C}_3$ and $\mathcal{C}_4$.*

Proof. As the procedure only ever adds nodes and edges that exist in G, it follows that G_i is a subgraph of G. In each iteration starting at line 3, we have from line 4 that if $n \in N_i$, then it is not contained in N_{us}. This means any goal node has outdegree 0 due to the initialization in line 1. Moreover, when n is added to G_i in line 7 it not belong to N_{us} in subsequent iterations. Therefore any n added to G_i has exactly outdegree 1, and so we have that G_i satisfies $\mathcal{C}_3$. Inspecting line 6 and 7, we have from $Solved(m, G, N')$ that when some m is added to G_i, each direct successor of m in G must already be contained in G_i. Therefore the operation is well-defined, and as every outgoing edge from m in G is added, it follows that G_i satisfies $\mathcal{C}_4$. □

Lemma A.8 (Extended proof Lemma 3.39)**.** *Given a saturated planning graph $G = (N, M, E, n_0)$ for $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \varphi_g)$, and let $G_i = (N_i, M_i, E_i)$. Further, let $n^* \in (N \setminus N_i)$ s.t. n^* has a direct successor m^* in G with Solved(m^*, G, N_i), and where $n_1, \ldots, n_y \in N_i$ denotes the direct successors of m^*. If for each x s.t. $1 \leq x \leq y$ there exists a solved subgraph of G_i for n_x, then there exists a solved subgraph of G for n^*.*

Proof. For each x s.t. $1 \leq x \leq y$ we let $H_x = (N_x, M_x, E_x, n_x)$ denote the solved subgraph of G_i for n_x. Further, let $N^* = N_1 \cup \cdots N_y \cup \{n^*\}$, $M^* = M_1 \cup \cdots M_y \cup \{m^*\}$ and $E^* = E_1 \cup \cdots E_y \cup \{(n^*, m^*), (m^*, n_1), \ldots, (m^*, n_y)\}$. We show that $G^* = (N^*, M^*, E^*, n^*)$ is a solved subgraph of G. See also Figure 3.5 for a more accommodating illustration of this construction.

Each H_x is by Lemma A.7 a subgraph of G, so from $n^* \in N$ and m^* being a direct successor of n^* in G, it follows that G^* is a subgraph of G. There is a path from n^* to m^*, and from m^* to each $n_1, \ldots, n_y$. As each H_x satisfies $\mathcal{C}_1$ this implies that every node in G^* is reachable from n^*. Since by construction n^* has indegree 0, it follows that G^* satisfies $\mathcal{C}_1$. Furthermore, by Lemma A.7 we have that each H_x satisfy $\mathcal{C}_3$ and $\mathcal{C}_4$. As the outdegree of n^* is 1, and m^* has exactly the same direct successors as in G, we therefore have that G^* satisfies $\mathcal{C}_3$ and $\mathcal{C}_4$.

For $\mathcal{C}_2$, assume towards a contradiction that G^* contains a cycle C. Due to the way we constructed G^* it is clear that n^* and m^* cannot be part of C. We can therefore assume that some node in C belongs to H_x for some $x \in \{1, \ldots, y\}$. By $\mathcal{C}_2$ we have that there is no cycle in H_x. Therefore C must visit another a node in $G_{x'}$ for some $x' \neq x$, and this node does not belong to H_x. This leads to either of the following two cases.

- There is some $(n_a, m_b) \in E^*$ s.t. $n_a \in N_x$ and $m_b \notin M_x$: As H_x satisfies $\mathcal{C}_3$, we either have $\mathcal{M}(n_a) \models \varphi_g$, or that there is some $(n_a, m_a) \in E_x$ with $m_a \neq m_b$. Since G^* satisfies $\mathcal{C}_3$ (shown above), the first case is contradicted by $(n_a, m_a) \in E^*$ because here n_a must be a sink; and the second case is contradicted by the outdegree of n_a in G^* being at least 2 since $\{(n_a, m_b), (n_a, m_a)\} \subseteq E^*$.
- There is some $(m_a, n_b) \in E^*$ s.t. $m_a \in M_x$ and $n_b \notin N_x$: Because $(m_a, n_b) \in E^*$ it must be that $(m_a, n_b) \in E_i$, as otherwise this edge could not be in G^*. As H_x is, by our initial assumption, a solved subgraph of G_i it follows from H_x satisfying $\mathcal{C}_4$ that $(m_a, n_b) \in E_x$, contradicting that $n_b \notin N_x$.

Having shown that G^* satisfies the necessary criteria the result follows. □

A.3 Proofs from Section 3.4

Lemma A.9 (Extended proof of Lemma 3.48)**.** *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $i = |\sigma_1| + 1 \leq p(n)$ being the position of T's head, and where c yields c' due to $\delta(q, s) = \{(q', s', d)\}$. Then $\mathsf{step}_{i,q,s}^{q',s',d}$ is applicable in $\mathcal{M}_c$, and $\mathcal{M}_c \otimes \mathsf{step}_{i,q,s}^{q',s',d}$ is the information cell associated with c'.*

Proof. This result is shown by considering the various cases of c and $\delta(q, s)$. Here we show one case; the remaining are similarly forthcoming. Let $x, y, z \in \Sigma$ and consider now that $\sigma_2(1) = x$. Further let $z = \sigma_2(2)$ and $\sigma_2' = \sigma_2(3) \dots \sigma_2(|\sigma_2|)$, and assume that $\delta(q, x) = \{(q', y, R)\}$. This means $c = \sigma_1 q x z \sigma_2'$ and $c' = \sigma_1 y q' z \sigma_2'$. In other words, T is in state q, reading x, and then changes state to q', writes y and moves right to read z.

By assumption $\mathcal{M}_c$ is the information cell associated with c, meaning that its single world satisfies q, h_i, x_i, z_{i+1}, the remaining letter-position symbols indicated by $\sigma_1 \sigma_2$, as well as blank symbols $\sqcup_j$ for $j > |\sigma_1| + |\sigma_2|$. Therefore $\mathcal{M}_c \models h_i \wedge q \wedge x_i$ and so $\mathsf{step}_{i,q,x}^{q',y,R}$ is applicable in $\mathcal{M}_c$. Consequently, $\mathcal{M}_c \otimes \mathsf{step}_{i,q,x}^{q',y,R}$ is a singleton, and by inspecting the boolean postconditions we see that this information cell satisfies q' (and not q), y_i (and not x_i) and h_{i+1} (and not h_i), with the remaining letter-position symbols (namely z_{i+1}) having the same truth value as in $\mathcal{M}_c$. Consequently $\mathcal{M}_c \otimes \mathsf{step}_{i,q,x}^{q',y,R}$ is isomorphic to the information cell associated with c'. □

Lemma A.10 (Extended proof of Lemma 3.65)**.** *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $|\sigma_1| + 1 \leq e(n)$, that yields c' due to $\delta(q, s) = \{(q', s', d)\}$. Then $\mathsf{estep}_{q,s}^{q',s',d}$ is applicable in $\mathcal{M}_c$, and $\mathcal{M}_c \otimes \mathsf{estep}_{q,s}^{q',s',d}$ is the information cell associated with c'.*

Proof. Let $x, z \in \Sigma$, and assume that $\sigma_2 = x$ and $\delta(q, x) = \{(q', z, R)\}$. We therefore have that $c = \sigma_1 q x$ and $c' = \sigma_1 z q' \sqcup$, and let $m = |\sigma_1| + 1$. This means that T is reading x in state q, writes z and moves right to read $\sqcup$. Let $\mathcal{M}_c$ be the information cell associated with c with valuation V. By definition $\mathcal{M}_c$contains worlds w_m and w_{m+1} s.t $w_m \in V(id_i)$ if the m_i bit is 1, and $w_{m+1} \in V(id_i)$ if the $(m+1)_i$ bit is 1. Further we have $w_m, w_{m+1} \in V(h_i)$ if the m_i bit is 1 (the head position is the same in every world). Because $m > |\sigma_1|$, $m \leq |\sigma_1| + 2$ and $\sigma_2(m - |\sigma_1|) = \sigma_2(1) = x$ we have $w_m \in V(x)$; and since $m + 1 > |\sigma_1| + |\sigma_2|$, $m + 1 \leq e(n) + 1$ we have $w_{m+1} \in V(\sqcup)$.

We now have that $\mathcal{M}_c, w_m \models (s \wedge (h = id))$, hence $\mathcal{M}_c \models \widehat{K}(s \wedge (h = id))$. The remaining worlds of $\mathcal{M}_c$ are given so that $\mathcal{M}_c \models init_b \wedge q \wedge (m \leq e(n)) \wedge$

$\widehat{K}(s \wedge (h = id))$ and therefore $\mathsf{estep}_{q,x}^{q',z,R}$ is applicable in $\mathcal{M}_c$. Furthermore, as $\mathsf{estep}_{q,x}^{q',z,R}$ is a singleton we have that $\mathcal{M}_c \otimes \mathsf{estep}_{q,x}^{q',z,R}$ contains the same number of worlds as $\mathcal{M}_c$. To show that the valuation of $\mathcal{M}_c \otimes \mathsf{estep}_{q_1,x}^{q_2,z,R}$ is identical to that of $\mathcal{M}_{c'}$, the cases for state and head symbols are immediate, and id is unchanged. We have $\mathcal{M}_c, w_m \not\models post(e)(x)$ and $\mathcal{M}_c, w_m \models post(e)(z)$, and so x does not hold in (w_m, e) whereas z does. We have that $\mathcal{M}_c, w_{m+1} \models \sqcup$, so from $post(e)(\sqcup) = \sqcup$ it follows that $\sqcup$ holds in (w_{m+1}, e) (note that we do not in general have $s \neq \sqcup$, so indeed blanks can be erased, though never written due to the definition of δ). Moreover, any letter symbol different from x and z retains its truth value, and so does x and z in worlds satisfying $h \neq id$; i.e. in every world distinct from w_m. From this it follows that $\mathcal{M}_c \otimes \mathsf{estep}_{q,x}^{q',z,R}$ is isomorphic to $\mathcal{M}_{c'}$. The remaining cases are similarly gruelling. □

Lemma A.11. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $|\sigma_1| + 1 \leq e(n)$, and where c yields c' due to $\delta(q, s) = \{(q', s', d)\}$. Then $\mathcal{M}_c \models \left[\!\left[\mathsf{estep}_{q,s}^{q',s',d}\right]\!\right]_s \varphi$ iff $\mathcal{M}_{c'} \models \varphi$.*

Proof. As in the proof of Lemma 3.49, except here using Lemma A.10 to show that $\mathcal{M}_c \models \langle \mathsf{estep}_{q,s}^{q',s',d} \rangle \top$ and $\mathcal{M}_c \otimes \mathsf{estep}_{q,s}^{q',s',d}$ is isomorphic to $\mathcal{M}_{c'}$. □

Proposition A.12 (Extended proof of Proposition 3.66)**.** *If the DTM T accepts σ without violating the space bound $e(n)$, then there exists a solution to the non-branching and partially observable planning problem $\mathcal{P}$.*

Proof. As the first order of business we show that if $c = \sigma_1 q \sigma_2$ is an x-accepting configuration with $|\sigma_1| + 1 \leq e(n)$, then there is a π s.t. to $\mathcal{M}_c \models [\![\pi]\!]_s \varphi_g$. To this end we proceed by induction on x. If c is 0-accepting then $l(q) = \mathsf{acc}$ and the head position is at most $e(n)$. It follows that $\mathcal{M}_c \models q \wedge (h \leq e(n))$ and therefore $\mathcal{M}_c \models [\![\mathsf{skip}]\!]_s \varphi_g$ thus completing the base case. Assume for the induction step that c is $(x+1)$-accepting. Then c yields c' due to some $\delta(q, s) = \{(q', s', d)\}$ and c' is y-accepting for some $y \leq x$. To see that the head position of c' is at most $e(n)$, we have from T not violating the space bound that if $|\sigma_1| + 1 = e(n)$ then $d = L$. From this and c' being y-accepting we can apply the induction hypothesis and conclude that $\mathcal{M}_{c'} \models [\![\pi]\!]_s \varphi_g$ for some π. Using Lemma A.11 it follows that $\mathcal{M}_c \models \left[\!\left[\mathsf{estep}_{q,s}^{q',s',d}\right]\!\right]_s ([\![\pi]\!]_s \varphi_g)$, hence $\mathcal{M}_c \models \left[\!\left[\mathsf{estep}_{q,s}^{q',s',d}; \pi\right]\!\right]_s \varphi_g$ as required.

Since $c_0 = q\sigma$ we can now conclude that there exists a plan π s.t. $\mathcal{M}_{c_0} \models [\![\pi]\!]_s \varphi_g$. Using Lemma 3.63 we have that $\mathcal{M}_s \models [\![\mathsf{boot}_0; \ldots; \mathsf{boot}_{b-1}; \mathsf{finalize}]\!]_s ([\![\pi]\!]_s \varphi_g)$, hence $\mathcal{M}_s \models [\![\mathsf{boot}_0; \ldots; \mathsf{boot}_{b-1}; \mathsf{finalize}; \pi]\!]_s \varphi_g$ as required. □

Lemma A.13. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $|\sigma_1| + 1 \leq e(n)$ and $s = \sigma_2(1)$. If there is an action $\mathcal{E} \in \mathsf{A}$ s.t. such that $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s \varphi$, then c yields*

some configuration c', and $\mathcal{M}_{c'} \models \varphi$.

Proof. From $\mathcal{M}_c$ being the information cell associated with c it follows that $\mathcal{M}_c \models init_b \wedge q \wedge (h = |\sigma_1|+1) \wedge \widehat{K}(s \wedge (h = id))$. This rules out $\mathcal{E}$ from being any bootstrapping action. As $\mathcal{E}$ exists and is applicable it follows from construction of the action library that there is some $(q', s', d) \in \delta(q, s)$. Therefore c has a successor configuration c'. Further we must have that $\mathcal{E} = \mathsf{estep}_{q,s}^{q',s',d}$ because this is the only action in A that is applicable in $\mathcal{M}_c$. By Lemma A.11 and $\mathcal{M}_c \models \left[\!\left[\mathsf{estep}_{q,s}^{q',s',d}\right]\!\right]_s \varphi$ it follows that $\mathcal{M}_{c'} \models \varphi$. □

Proposition A.14 (Extended proof of Proposition 3.67)**.** *If there exists a solution to the non-branching and partially observable planning problem $\mathcal{P}$, then the DTM T accepts σ without violating the space bound $e(n)$.*

Proof. In order to show this, we first argue that the bootstrapping procedure always takes place. To this end consider some information cell $\mathcal{M}$. We have for $0 \leq i \leq b-1$, that if $init_i$ holds and no state symbol holds, the only applicable action in $\mathcal{M}$ is boot_i, and when $init_b$ holds and no state symbol holds, the only applicable action in $\mathcal{M}$ is $\mathsf{finalize}$. Furthermore φ_g requires that *some* (accepting) state symbol hold, and thus we must have $\mathsf{finalize}$ as part of any solution. Therefore we can assume that any solution to $\mathcal{P}$ has the prefix $\mathsf{boot}_0; \ldots; \mathsf{boot}_{b-1}; \mathsf{finalize}$. From Lemma 3.63 we have that when $\mathcal{M}_{c_0} \models \varphi$ then $\mathcal{M}_s \models [\![\mathsf{boot}_0; \ldots; \mathsf{boot}_{b-1}; \mathsf{finalize}]\!]_s \varphi$. Consequently we're done after showing that if there exists a π s.t. $\mathcal{M}_{c_0} \models [\![\pi]\!]_s \varphi_g$, then c_0 is an accepting configuration and T does not violate its space bound.

Letting $c = \sigma_1 q \sigma_2$ be a configuration of T s.t. $|\sigma_1| + 1 \leq e(n)$, we will show the stronger result that if $\mathcal{M}_c \models [\![\pi]\!]_s \varphi_g$, then c is accepting and no further computation violates the space bound. We proceed by induction on the length of π (cf. Theorem 2.25), with the cases of $\pi = \mathsf{skip}$ and $\pi = \mathsf{if}\ \varphi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ being as in the proof of Proposition 3.52. For the second base case $\pi = \mathcal{E}$ for some $\mathcal{E} \in \mathsf{A}$, assume that $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s \varphi_g$. By Lemma A.13 we have that c yields a configuration c', and that $\mathcal{M}_{c'} \models \varphi_g$. Therefore c' is 0-accepting and c is 1-accepting. For the space bound to be violated the head position in c' must be $e(n)+1$, as by assumption the head position in c is at most $e(n)$. But since $\mathcal{M}_{c'} \models (h \leq e(n))$ we derive the contradiction $\mathcal{M}_{c'} \models (h = (e(n)+1)) \wedge (h \leq e(n))$, thus concluding the base case.

For the induction step what remains is to consider $\pi = \pi_1; \pi_2$ with $|\pi| = m+1$, where we assume π has the form $\mathcal{E}; \pi'$ with $|\pi'| \leq m$ (cf. Proposition 3.52). Assuming that $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s ([\![\pi']\!]_s \varphi_g)$, it follows from Lemma A.13 that c yields a configuration c', and that $\mathcal{M}_{c'} \models [\![\pi']\!]_s \varphi_g$. By assuming the head position in c' is $e(n)+1$, no actions are applicable in $\mathcal{M}_{c'}$ and so we can derive the

contradiction $\mathcal{M}_{c'} \models (h = (e(n)+1) \wedge ([\![\mathsf{skip}]\!]_s\,(h \leq e(n)))$. We can therefore apply the induction hypothesis and conclude that c' is x-accepting and no further computation from c' violates the space bound, hence c is $(x+1)$-accepting and no further computation from c violates the space bound. □

Lemma A.15. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $|\sigma_1| + 1 \leq e(n)$ (the head position), and where c has successor configurations $c'_1, \ldots, c'_k$ due to $\delta(q,s) = \{(q^1,s^1,d^1),\ldots,(q^k,s^k,d^k)\}$.*

- *If $l(q) = \exists$ then for each $j \in \{1,\ldots,k\}$ we have that $\mathsf{estep}^{q^j,s^j,d^j}_{q,s}$ is applicable in $\mathcal{M}_c$, and $\mathcal{M}_c \otimes \mathsf{estep}^{q^j,s^j,d^j}_{q,s}$ is the information cell associated with c'_j, and*
- *If $l(q) = \forall$ then $\mathsf{estep}^{\forall}_{q,s}$ is applicable in $\mathcal{M}_c$, and $\mathcal{M}_c \otimes \mathsf{estep}^{\forall}_{i,q,s}$ is the disjoint union of the information cells associated with $c'_1, \ldots, c'_k$.*

Proof. Using Lemma A.10 this is as in the proof of Lemma 3.54. □

Lemma A.16. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $|\sigma_1| + 1 \leq e(n)$, $l(q) = \exists$, and where c has successor configurations $c'_1, \ldots, c'_k$ due to $\delta(q,s) = \{(q^1,s^1,d^1),\ldots,(q^k,s^k,d^k)\}$. Then for each $j \in \{1,\ldots,k\}$ we have that $\mathcal{M}_c \models [\![\mathsf{estep}^{q^j,s^j,d^j}_{q,s}]\!]_s\,\varphi$ iff $\mathcal{M}_{c'_j} \models \varphi$.*

Proof. As in the proof of Lemma A.11, here by using Lemma A.15 rather than Lemma A.10. □

Lemma A.17 (Extended proof of Lemma 3.69)**.** *For the definition of ψ_j, we refer to the discussion above Lemma 3.69. Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $|\sigma_1|+1 \leq e(n)$, $l(q) = \forall$, and where c has successor configurations $c'_1, \ldots, c'_k$ due to $\delta(q,s) = \{(q^1,s^1,d^1),\ldots,(q^k,s^k,d^k)\}$. If $\mathcal{M}_{c'_j} \models [\![\pi_j]\!]_s\,\varphi$ for any $j \in \{1,\ldots,k\}$, then $\mathcal{M}_c \models [\![\mathsf{estep}^{\forall}_{i,q,s}; \mathsf{if}\ \psi_1\ \mathsf{then}\ \pi_1\ \mathsf{else\ if}\ \cdots\ \mathsf{if}\ \psi_k\ \mathsf{then}\ \pi_k]\!]_s\,\varphi$.*

Proof. As in the proof of Lemma 3.56, we have for any $j \in \{1,\ldots,k\}$ that $\mathcal{M}_{c'_j} \models ((\psi_1 \to [\![\pi_1]\!]_s\,\varphi) \wedge \cdots \wedge (\psi_j \to [\![\pi_j]\!]_s\,\varphi)) \wedge (\psi_1 \vee \cdots \vee \psi_k)$. Again using the derivation in the proof of Theorem 2.24 it follows that

$$\mathcal{M}_{c'_j} \models K\,[\![\mathsf{if}\ \psi_1\ \mathsf{then}\ \pi_1\ \mathsf{else\ if}\ \cdots\ \mathsf{if}\ \psi_k\ \mathsf{then}\ \pi_k]\!]_s\,\varphi$$
$$\mathcal{M}_c \models [\mathsf{estep}^{\forall}_{q,s}]\,(K\,[\![\mathsf{if}\ \psi_1\ \mathsf{then}\ \pi_1\ \mathsf{else\ if}\ \cdots\ \mathsf{if}\ \psi_k\ \mathsf{then}\ \pi_k]\!]_s\,\varphi)$$

because $\mathcal{M}_c$ is the disjoint union of each $\mathcal{M}_{c'_1}, \ldots, \mathcal{M}_{c'_k}$ by Lemma A.15 $\mathcal{M}_c \otimes \mathsf{estep}^{\forall}_{q,s}$. As $\mathsf{step}^{\forall}_{q,s}$ is applicable in $\mathcal{M}_c$ we conclude as required:

$$\mathcal{M}_c \models [\![\mathsf{estep}^{\forall}_{q,s}; \mathsf{if}\ \psi_1\ \mathsf{then}\ \pi_1\ \mathsf{else\ if}\ \cdots\ \mathsf{if}\ \psi_k\ \mathsf{then}\ \pi_k]\!]_s\,\varphi$$

□

Proposition A.18 (Extended proof of Proposition 3.70)**.** *If the ATM T accepts σ without violating the space bound $e(n)$, then there exists a solution to the branching and partially observable planning problem $\mathcal{P}$.*

Proof. We show that if T is in a configuration $c = \sigma_1 q \sigma_2$ which is x-accepting and $|\sigma_1| + 1 \leq e(n)$, then there is a plan π s.t. $\mathcal{M}_c \models [\![\pi]\!]_s \, \varphi_g$, where $\mathcal{M}_c$ is the information cell associated with c. The proof is by induction on x and is almost exactly as the proof of Proposition 3.57, with two minor changes. First we use Lemmas A.16 and A.17 for the induction step (instead of Lemmas 3.55 and 3.56). Second the space bound here is $e(n)$ rather than $p(n)$. As $c_0 = q_0 \sigma$ is such a configuration it follows that there is a π s.t. $\mathcal{M}_{c_0} \models [\![\pi]\!]_s \, \varphi$, and so applying Lemma 3.63 we conclude that $\mathcal{M}_s \models [\![\mathsf{boot}_0; \ldots; \mathsf{boot}_{b-1}; \mathsf{finalize}; \pi]\!]_s \, \varphi_g$. □

Lemma A.19. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $|\sigma_1| + 1 \leq e(n)$, $s = \sigma_2(1)$ and $l(q) = \exists$. If there is an action $\mathcal{E} \in \mathsf{A}$ such that $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s \, \varphi$, then c has a successor configuration c', and $\mathcal{M}_{c'} \models \varphi$.*

Proof. As in the proof of Lemma A.13 we have that $\mathcal{E}$ cannot be a bootstrapping action, and moreover as $\mathcal{E}$ is applicable in $\mathcal{M}_c$ we have that $\delta(q, s) = \{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$. Consequently we have for any $j \in \{1, \ldots, k\}$ that there exists a successor configuration c'_j. Further, we may assume $\mathcal{E} = \mathsf{estep}^{q^j, s^j, j}_{q,s}$ so using Lemma A.16 and $\mathcal{M}_c \models \left[\!\left[\mathsf{estep}^{q^j, s^j, j}_{q,s}\right]\!\right]_s \varphi$ we conclude $\mathcal{M}_{c'_j} \models \varphi$. □

Lemma A.20. *Let $c = \sigma_1 q \sigma_2$ be a configuration of T with $|\sigma_1| + 1 \leq e(n)$, $s = \sigma_2(1)$ and $l(q) = \forall$. If there is an action $\mathcal{E} \in \mathsf{A}$ s.t. such that $\mathcal{M}_c \models [\![\mathcal{E}]\!]_s \, \varphi$, then c has successor configurations $c'_1, \ldots, c'_k$, and $\mathcal{M}_{c'_j} \models \varphi$ for each $j \in \{1, \ldots, k\}$.*

Proof. From our assumptions the only applicable action in $\mathcal{M}_c$ is $\mathsf{estep}^{\forall}_{q,s}$. We therefore have $\delta(q, s) = \{(q^1, s^1, d^1), \ldots, (q^k, s^k, d^k)\}$, hence c has successor configurations $c'_1, \ldots, c'_k$. By Lemma A.15 we have that $\mathcal{M}_c \otimes \mathsf{estep}^{\forall}_{q,s}$ is the disjoint union of each $\mathcal{M}_{c'_1}, \ldots, \mathcal{M}_{c'_k}$, and so as $\mathcal{M}_c \otimes \mathsf{estep}^{\forall}_{q,s} \models \varphi$ it follows that $\mathcal{M}_{c'_j} \models \varphi$ for each $j \in \{1, \ldots, k\}$. □

Proposition A.21 (Extended proof of Proposition 3.71)**.** *If there exists a solution to the branching and partially observable planning problem $\mathcal{P}$, then the ATM T accepts σ without violating the space bound $e(n)$.*

Proof. As in the proof of Proposition A.14 we have that the bootstrapping procedure always takes place. What must therefore be shown is that, if there exists a π s.t. $\mathcal{M}_{c_0} \models [\![\pi]\!]_s \, \varphi_g$, then c_0 is an accepting configuration and T

does not violate its space bound. Letting $c = \sigma_1 q \sigma_2$ be a configuration of T s.t. $|\sigma_1| + 1 \leq e(n)$, we can show the stronger result that if $\mathcal{M}_c \models [\![\pi]\!]_s \, \varphi_g$, then c is accepting and no further computation violates the space bound. When the prefix of π is $\mathcal{E}$ we apply either Lemma A.19to show that an existential configuration is accepting, or Lemma A.20 to show that a universal configuration is accepting. That the space bound is not violated is shown by an indirect proof, which is essentially that used in the proof of Proposition A.14. The case of $\pi = \mathsf{skip}$ is immediate, and $\pi = \mathsf{if}\ \varphi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ follows straight from the induction hypothesis. □

Bibliography

[Ågotnes et al., 2012] Ågotnes, T., van der Hoek, W., and Wooldridge, M. (2012). Conservative social laws. In Raedt, L. D., Bessière, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., and Lucas, P. J. F., editors, *ECAI*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 49–54. IOS Press.

[Alur et al., 2002] Alur, R., Henzinger, T. A., and Kupferman, O. (2002). Alternating-time temporal logic. *J. ACM*, 49(5):672–713.

[Andersen et al., 2012] Andersen, M., Bolander, T., and Jensen, M. (2012). Conditional epistemic planning. In *Proc. of 13th JELIA*, LNCS 7519, pages 94–106. Springer.

[Andersen et al., 2014] Andersen, M., Bolander, T., and Jensen, M. (2014). Don't plan for the unexpected: Planning based on plausibility models (to appear). *Logique et Analyse*.

[Andersen et al., 2013] Andersen, M. B., Bolander, T., van Ditmarsch, H. P., and Jensen, M. H. (2013). Bisimulation for single-agent plausibility models. In Cranefield, S. and Nayak, A., editors, *Australasian Conference on Artificial Intelligence*, volume 8272 of *Lecture Notes in Computer Science*, pages 277–288. Springer.

[Aucher, 2005] Aucher, G. (2005). A combined system for update logic and belief revision. In *Proc. of 7th PRIMA*, pages 1–17. Springer. LNAI 3371.

[Aucher, 2010] Aucher, G. (2010). An internal version of epistemic logic. *Studia Logica*, 94(1):1–22.

[Aucher, 2012] Aucher, G. (2012). Del-sequents for regression and epistemic planning. *Journal of Applied Non-Classical Logics*, 22(4):337–367.

[Aucher and Bolander, 2013] Aucher, G. and Bolander, T. (2013). Undecidability in Epistemic Planning. Rapport de recherche.

[Aucher and Schwarzentruber, 2012] Aucher, G. and Schwarzentruber, F. (2012). On the Complexity of Dynamic Epistemic Logic (Extended Version). Rapport de recherche RR-8164, INRIA.

[Aumann and Hart, 1992] Aumann, R. and Hart, S., editors (1992). *Handbook of Game Theory with Economic Applications*. Elsevier.

[Baltag et al., 1998] Baltag, A., Moss, L., and Solecki, S. (1998). The logic of public announcements, common knowledge, and private suspicions. In *Proc. of 7th TARK*, pages 43–56.

[Baltag and Moss, 2004] Baltag, A. and Moss, L. S. (2004). Logics for Epistemic Programs. *Synthese*, 139:165–224.

[Baltag and Smets, 2006] Baltag, A. and Smets, S. (2006). Dynamic belief revision over multi-agent plausibility models.

[Baltag and Smets, 2008] Baltag, A. and Smets, S. (2008). A qualitative theory of dynamic interactive belief revision. In *Proc. of 7th LOFT*, Texts in Logic and Games 3, pages 13–60. Amsterdam University Press.

[Baral et al., 2012] Baral, C., Gelfond, G., Pontelli, E., and Son, T. C. (2012). An action language for reasoning about beliefs in multi-agent domains. In *Proceedings of the 14th International Workshop on Non-Monotonic Reasoning*.

[Baron-Cohen et al., 1985] Baron-Cohen, S., Leslie, A., and Frith, U. (1985). Does the autistic child have a 'theory of mind'? *Cognition*, 21:37–46.

[Barwise and Moss, 1996] Barwise, J. and Moss, L. (1996). *Vicious circles*. CSLI Publications.

[Belnap et al., 2001] Belnap, N., Perloff, M., and Xu, M. (2001). *Facing the future : Agents and choices in our indeterminist world*. Oxford University Press.

[Benthem and Pacuit, 2013] Benthem, J. v. and Pacuit, E. (2013). Connecting logics of choice and change.

[Bentzen, 2010] Bentzen, M. M. (2010). *Stit, Iit, and Deontic Logic for Action Types*. PhD thesis, Department of Philosophy, Roskilde University.

[Bertoli et al., 2001] Bertoli, P., Cimatti, A., and Roveri, M. (2001). Conditional planning under partial observability as heuristic-symbolic search in belief space. In *ECP-01. Sixth European Conference on Planning, Toledo, Spain*, pages 379–384.

[Bicks, 2010] Bicks, M. (2010). Artificial Intelligence Pioneer. http://www.pbs.org/wgbh/nova/tech/pioneer-artificial-intelligence.html. [Online; accessed 02-October-2013].

[Blackburn et al., 2001] Blackburn, P., de Rijke, M., and Venema, Y. (2001). *Modal Logic*. Cambridge University Press, Cambridge. Cambridge Tracts in Theoretical Computer Science 53.

[Blackburn and van Benthem, 2006] Blackburn, P. and van Benthem, J. (2006). Modal logic: A semantic perspective. In *Handbook of Modal Logic*. Elsevier.

[Blackburn et al., 2006] Blackburn, P., van Benthem, J., and Wolter, F., editors (2006). *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*. Elsevier Science.

[Bloom and German, 2000] Bloom, P. and German, T. (2000). Two reasons to abandon the false belief task as a test of theory of mind. *Cognition*, 77(1):25–31.

[Bolander and Andersen, 2011] Bolander, T. and Andersen, M. B. (2011). Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34.

[Brenner and Nebel, 2009] Brenner, M. and Nebel, B. (2009). Continual planning and acting in dynamic multiagent environments. *Autonomous Agents and Multi-Agent Systems*, 19(3):297–331.

[Britz and Varzinczak, 2013] Britz, K. and Varzinczak, I. (2013). Defeasible modalities. In *Proc. of the 14th TARK*.

[Broersen et al., 2006a] Broersen, J., Herzig, A., and Troquard, N. (2006a). Embedding Alternating-time Temporal Logic in Strategic STIT Logic of Agency. *Journal of Logic and Computation*, 16(5):559–578.

[Broersen et al., 2006b] Broersen, J., Herzig, A., and Troquard, N. (2006b). From coalition logic to stit. *Electron. Notes Theor. Comput. Sci.*, 157(4):23–35.

[Bylander, 1994] Bylander, T. (1994). The computational complexity of propositional strips planning. *Artificial Intelligence*, 69:165–204.

[Chandra et al., 1981] Chandra, A., Kozen, D., and Stockmeyer, L. (1981). Alternation. *Journal of the ACM*, 28:114–33.

[Cimatti et al., 2003] Cimatti, A., Pistore, M., Roveri, M., and Traverso, P. (2003). Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 147(1-2):35–84.

[Condon, 1992] Condon, A. (1992). The complexity of stochastic games. *Information and Computation*, 96:203–224.

[de Lima, 2007] de Lima, T. (2007). *Optimal Methods for Reasoning about Actions and Plans in Multi-Agents Systems.* PhD thesis, IRIT, University of Toulouse 3, France.

[Demey, 2011] Demey, L. (2011). Some remarks on the model theory of epistemic plausibility models. *Journal of Applied Non-Classical Logics*, 21(3-4):375–395.

[Ditmarsch et al., 2007] Ditmarsch, H. v., van der Hoek, W., and Kooi, B. (2007). *Dynamic Epistemic Logic.* Springer.

[Fagin et al., 1995] Fagin, R., Halpern, J., Moses, Y., and Vardi, M. (1995). *Reasoning about Knowledge.* MIT Press, Cambridge MA.

[Fikes and Nilsson, 1971] Fikes, R. E. and Nilsson, N. J. (1971). Strips: A new approach to the application of theorem proving to problem solving. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, IJCAI'71, pages 608–620, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA.

[Ghallab et al., 1998] Ghallab, M., Howe, A., Knoblock, C., Mcdermott, D., Ram, A., Veloso, M., Weld, D., and Wilkins, D. (1998). PDDL—The Planning Domain Definition Language.

[Ghallab et al., 2004] Ghallab, M., Nau, D. S., and Traverso, P. (2004). *Automated Planning: Theory and Practice.* Morgan Kaufmann.

[Goranko and Jamroga, 2004] Goranko, V. and Jamroga, W. (2004). Comparing semantics of logics for multi-agent systems. *SYNTHESE*, 139(2):241–280.

[Grove, 1988] Grove, A. (1988). Two modellings for theory change. *Journal of Philosophical Logic*, 17:157–170.

[Haslum and Jonsson, 1999] Haslum, P. and Jonsson, P. (1999). Some results on the complexity of planning with incomplete information. In Biundo, S. and Fox, M., editors, *ECP*, volume 1809 of *Lecture Notes in Computer Science*, pages 308–318. Springer.

[Helmert, 2006] Helmert, M. (2006). The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246.

[Herzig et al., 2003] Herzig, A., Lang, J., and Marquis, P. (2003). Action representation and partially observable planning using epistemic logic. In Gottlob, G. and Walsh, T., editors, *IJCAI*, pages 1067–1072. Morgan Kaufmann.

[Herzig and Lorini, 2010] 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.

[Hintikka, 1962] Hintikka, J. (1962). *Knowledge and Belief*. Cornell University Press, Ithaca, NY.

[Horrocks et al., 2006] Horrocks, I., Hustadt, U., Sattler, U., and Schmidt, R. (2006). Computational modal logic. In *Handbook of Modal Logic*. Elsevier.

[JAMES, 2014] JAMES (2014). http://homepages.inf.ed.ac.uk/rpetrick/projects/james/. [Online; accessed 12-January-2014].

[Jamroga and Ågotnes, 2007] Jamroga, W. and Ågotnes, T. (2007). Constructive knowledge: what agents can achieve under imperfect information. *Journal of Applied Non-Classical Logics*, 17(4):423–475.

[Jensen, 2013a] Jensen, M. H. (2013a). The computational complexity of single agent epistemic planning (manuscript).

[Jensen, 2013b] Jensen, M. H. (2013b). Planning using dynamic epistemic logic: Correspondence and complexity. In Grossi, D., Roy, O., and Huang, H., editors, *LORI*, volume 8196 of *Lecture Notes in Computer Science*, pages 316–320. Springer.

[Kraus et al., 1990] Kraus, S., Lehmann, D., and Magidor, M. (1990). Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207.

[Lang and Zanuttini, 2013] Lang, J. and Zanuttini, B. (2013). Knowledge-based programs as plans: Succinctness and the complexity of plan existence. *in Proc. TARK 2013*.

[Laverny and Lang, 2004] Laverny, N. and Lang, J. (2004). From knowledge-based programs to graded belief-based programs part i: On-line reasoning. In de Mántaras, R. L. and Saitta, L., editors, *ECAI*, pages 368–372. IOS Press.

[Laverny and Lang, 2005] Laverny, N. and Lang, J. (2005). From knowledge-based programs to graded belief-based programs, part ii: off-line reasoning. In Kaelbling, L. P. and Saffiotti, A., editors, *IJCAI*, pages 497–502. Professional Book Center.

[Lewis, 1973] Lewis, D. (1973). *Counterfactuals*. Harvard University Press, Cambridge (MA).

[Littman, 1997] Littman, M. L. (1997). Probabilistic propositional planning: Representations and complexity. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, pages 748–754. AAAI Press.

[Löwe et al., 2011] Löwe, B., Pacuit, E., and Witzel, A. (2011). Del planning and some tractable cases. In *Proceedings of the Third International Conference on Logic, Rationality, and Interaction*, LORI'11, pages 179–192, Berlin, Heidelberg. Springer-Verlag.

[Meyer et al., 2000] Meyer, T., Labuschagne, W., and Heidema, J. (2000). Refined epistemic entrenchment. *Journal of Logic, Language, and Information*, 9:237–259.

[Pacuit, 2007] Pacuit, E. (2007). Some comments on history based structures. *J. Applied Logic*, 5(4):613–624.

[Parikh and Ramanujam, 2003] Parikh, R. and Ramanujam, R. (2003). A knowledge based semantics of messages. *Journal of Logic, Language and Information*, 12:453–467.

[Pauly, 2002] Pauly, M. (2002). A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166.

[Petrick and Bacchus, 2002] Petrick, R. P. A. and Bacchus, F. (2002). A knowledge-based approach to planning with incomplete information and sensing. In Ghallab, M., Hertzberg, J., and Traverso, P., editors, *AIPS*, pages 212–222. AAAI.

[Petrick and Bacchus, 2004] Petrick, R. P. A. and Bacchus, F. (2004). Extending the knowledge-based approach to planning with incomplete information and sensing. In [Zilberstein et al., 2004], pages 2–11.

[Premack and Woodruff, 1978] Premack, D. and Woodruff, G. (1978). Does the chimpanzee have a 'theory of mind'? *Behavioral and Brain Sciences*, 4:515–526.

[Rintanen, 2004] Rintanen, J. (2004). Complexity of planning with partial observability. In [Zilberstein et al., 2004], pages 345–354.

[Rintanen, 2005] Rintanen, J. (2005). Automated planning: Algorithms and complexity. Habilitation thesis, Albert-Ludwigs-Universität Freiburg.

[Schwarzentruber and Herzig, 2008] Schwarzentruber, F. and Herzig, A. (2008). Properties of logics of individual and group agency. *Advances in Modal Logic 2006*, 7:133–149.

[Segerberg, 1998] Segerberg, K. (1998). Irrevocable belief revision in dynamic doxastic logic. *Notre Dame Journal of Formal Logic*, 39(3):287–306.

[Sipser, 2006] Sipser, M. (2006). *Introduction to the theory of computation*. Thomson Course Technology, Boston, Massachusetts, 2 edition.

[Spohn, 1988] Spohn, W. (1988). Ordinal conditional functions: a dynamic theory of epistemic states. In Harper, W. and Skyrms, B., editors, *Causation in Decision, Belief Change, and Statistics*, volume II, pages 105–134.

[Stalnaker, 1996] Stalnaker, R. (1996). Knowledge, belief and counterfactual reasoning in games. *Economics and Philosophy*, 12:133–163.

[van Benthem, 1998] van Benthem, J. (1998). Dynamic odds and ends. Technical Report ML-1998-08, University of Amsterdam.

[van Benthem, 2001] van Benthem, J. (2001). Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–48.

[van Benthem, 2007] van Benthem, J. (2007). Dynamic logic of belief revision. *Journal of Applied Non-Classical Logics*, 17(2):129–155.

[van Benthem, 2011] van Benthem, J. (2011). *Logical Dynamics of Information and Interaction*. Cambridge University Press.

[van Benthem et al., 2007] van Benthem, J., Gerbrandy, J., and Pacuit, E. (2007). Merging frameworks for interaction: Del and etl. In *Proceedings of the 11th conference on Theoretical aspects of rationality and knowledge*, TARK '07, pages 72–81, New York, NY, USA. ACM.

[van der Hoek and Wooldridge, 2002] van der Hoek, W. and Wooldridge, M. (2002). Tractable multiagent planning for epistemic goals. In *Proc. of the First International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002)*, pages 1167–1174. ACM.

[van Ditmarsch, 2005] van Ditmarsch, H. (2005). Prolegomena to dynamic logic for belief revision. *Synthese (Knowledge, Rationality & Action)*, 147:229–275.

[van Ditmarsch, pear] van Ditmarsch, H. (2013, to appear). Revocable belief revision. *Studia Logica*.

[van Ditmarsch and Kooi, 2008] van Ditmarsch, H. and Kooi, B. (2008). Semantic results for ontic and epistemic change. In Bonanno, G., van der Hoek, W., and Wooldridge, M., editors, *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, Texts in Logic and Games, pages 87–117. Amsterdam University Press.

[van Ditmarsch and Labuschagne, 2007] van Ditmarsch, H. and Labuschagne, W. (2007). My beliefs about your beliefs – a case study in theory of mind and epistemic logic. *Synthese*, 155:191–209.

[van Ditmarsch et al., 2007] van Ditmarsch, H., van der Hoek, W., and Kooi, B. (2007). *Dynamic Epistemic Logic*. Springer Publishing Company, Incorporated, 1st edition.

[Yu et al., 2013] Yu, Q., Wen, X., and Liu, Y. (2013). Multi-agent epistemic explanatory diagnosis via reasoning about actions. In Rossi, F., editor, *IJCAI*. IJCAI/AAAI.

[Zilberstein et al., 2004] Zilberstein, S., Koehler, J., and Koenig, S., editors (2004). *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004), June 3-7 2004, Whistler, British Columbia, Canada*. AAAI.