Belief Tracking For Multi-agent Planning

Filippos Kominis

TESI DOCTORAL UPF / 2017

Thesis advisor

Prof. Dr. Hector Geffner,
Department of Information and Communication Technologies

To my family

Acknowledgements

This thesis was made possible with the help and support of many people. First of all, I would like to thank my PhD advisor Hector Geffner. Without his help, guidance and patience, these pages would not exist. His approach to tackling problems and his high expectations of what ”work”should look like can only be surpassed by his passion for research. He is a model of exceptional researcher, and I can only feel privileged to have been his student.

I would like to thank the people of the Artificial Intelligence group at UPF. Nir Lipovetzsky, Alex Albore and Hector Palacios, for making me feel welcome when I arrived, and for helping me get acquainted with the university, the group, the work, the people in conferences, and Barcelona itself. I also thank Anders Johnson, Vicenc Gomez, Gergely Neu and Dimitri Ognibene who, along with Hector, made the group meetings so enjoyable and educating. The DTIC administration employees deserve a thank you as well. Their experience and patience with people who don’t like deadlines was put to the test more than once. I am sure Lydia will agree.

These five years were filled with strong feelings. All-nighters trying to meet deadlines, frustration when things didn’t work, excitement when things did, discussions whenever was needed. All the ups and downs were shared with (and equally endured by) my fellow PhD candidates: Javier Segovia-Aguas, Guillem Frances, Oussam Larkem, Srinibas Swain. With some of them I shared many (many) coffees, with others trips to conferences and competitions. I am happy to have shared this experience with you. I would also like to thank all the people I met at the Dagstuhl seminar on epistemic planning. Specifically, Tristian Charrier and Francois Schwarzentruber, for their help concerning the epistemic logic part of the thesis.

While in Barcelona, I met people who will always be in my heart. Jonatan Ferrer (who tried hard to acquaint me with the catalan way of living), Bruno Paun (who was always there to help, give advise and make me laugh) and Damir Lotinac, with whom I spent these five years, from start to finish (and all in-between). I cannot imagine these years with somebody else by my side.

I owe a lot, if not all, to Niko Karaisko and Alexandro Goula. The good things I see in me should be attributed to you, and it is only fair to take full responsibility for the bad.

My deepest gratitude goes to Kalliopi Papagianni. Her support, love and patience were invaluable. I can only hope that the years to come will allow me to repay her in kind. I

am sure she is as excited as I am.

Finally, I would like to thank my parents and my sister. There are no words to express my gratitude for their support, their wisdom, their kindness and the lessons they tried to teach me. They were always there for me, respecting my choices and guiding me through life, each one with their own strength and perspective. What more to ask.

Abstract

Classical planning is the problem of finding a sequence of actions that achieve a desired goal from an initial state, assuming deterministic actions. Dynamic epistemic logic (DEL) on the other hand, provides formal frameworks that allow the modeling of complex beliefs in multi-agent settings and define how those beliefs change due to physical and communication actions. In this dissertation we focus on bridging the gap between the expressivity of DEL and the computational approaches used in classical planning. First, we present formulations that capture a fragment of the expressivity of DEL and can model nested knowledge in two different multi-agent settings. Second, we tackle the computational problem of finding plans by providing translations to classical planning that allow the use of classical planners and heuristic search. We empirically evaluate our approaches and discuss their formal properties.

Resumen

La Planificació Clàssica és un problema que busca una seqüència d'accions per arribar a una meta o objectiu des d'un estat inicial, assumint que les accions són deterministes. Per una altra banda, la Lògica Epistèmica Dinàmica (LED), proporciona una eina de treball formal que permet el modelatge de creences complexes en un entorn de múltiples agents, i defineix com aquestes creences varien aplicant accions físiques i de comunicació. En aquesta disertació ens centrem en connectar l'expressivitat de LED amb els diferents enfocaments que s'utilitzen a planificació clàssica. Primer presentem les formulacions que capturen un fragment de l'expressivitat de LED i que pot modelar coneixement anidat en dos configuracions diferents amb múltiples agents. Després abordem el problema computacional de trobar plans, tot proporcionant traduccions de planificació clàssica que permeten utilitzar planificadors clàssics amb búsquedes heurístiques. Finalment, evaluem de forma empírica els nostres enfocaments i parlem sobre les seves propietats formals.

Preface

Classical planning is the problem of finding a sequence of actions that achieves a goal, given an initial situation. It is the simplest problem of automated planning, since it assumes complete knowledge of the environment and deterministic actions. The problem can be mapped to a path finding problem, where nodes represent states (the values of a set of variables) and directed edges represent actions that allow the transition from one state to the next.

In the worst case, determining if a given classical planning instance in STRIPS has a solution is PSPACE-complete (Bylander, 1994). Despite the worst-case complexity, state-of-the-art classical planners (which take as input a compact representation of the problem) efficiently solve a variety of problems, as demonstrated by their performance on benchmarks from planning competitions [1]. This was achieved by devising a number of search approaches, such as heuristic search (Bonet et al., 1997; McDermott, 1996), where the search for a solution is guided by a function which estimates the distance (in terms of action cost) of a state to a goal state, and width-based search (Lipovetzky and Geffner, 2012), where states are not evaluated based on their distance to the goal but on their novelty. Further efficiency has been achieved with search enhancements such as helpful actions and landmarks (Hoffmann and Nebel, 2001; Richter and Westphal, 2010).

Classical planning predominantly refers to single-agent planning: an agent that acts within an environment in order to achieve his goal. In the presence of other agents, planning must take into consideration a number of additional factors: are the agents collaborating or not, do they need to coordinate, must they reach an agreement about resources etc. We are interested in a specific aspect of multi-agent planning which is the ability to plan by taking into consideration the knowledge the agents have about (i) the environment and how their actions affect it, and (ii) the knowledge the other agents' have and how it is affected by actions and communication.

Dynamic Epistemic Logic (DEL) (Van Benthem, 2011) focuses on such issues: how to reason about knowledge and beliefs in a dynamic, multi-agent setting. DEL is an umbrella term for a number of different logics that study how actions and communication affect knowledge and beliefs. Based on Kripke models and the notions of *state* and *indistinguishability* (Kripke, 1971), DEL approaches provide formal frameworks for modeling complex beliefs in multi-agent situations and how they change due to physical and information-sharing actions.

[1] http://www.icaps-conference.org/index.php/Main/Competitions

Classical planning focuses on computationally efficient approaches for solving planning problems, while DEL investigates the formal semantics and expressivity of theories about knowledge/beliefs and *change*. In this dissertation we focus on bridging the gap between them by providing formulations that (i) capture *part* of the expressivity of DEL in terms of high-order (nested) knowledge, and (ii) allow, through translation approaches, the use of classical planners and heuristic search to find plans.

In Part I of the dissertation we review the necessary background. We start with the classical, conformant and contingent planning models. The classical planning model is relevant to the translations we present in later chapters, the conformant and contingent planning models are relevant to the problems we have and their type of solutions. Then we present a categorization of multi-agent planning problems and, lastly, we review (Dynamic) Epistemic Logic, in terms of language and semantics.

In Part II we introduce a belief representation $B(t)$, which is the starting point for the rest of the work in this dissertation. In the first chapter of this part, the belief representation $B(t)$ is used for modeling linear, multi-agent planning problems, where a plan is a sequence of actions that must achieve the goal for all possible initial states. It allows for nested epistemic literals to appear in goals, preconditions and as parameters in sensing and update actions. We present the dynamics of our formulation by specifying how different types of actions change the belief representation $B(t)$, and how $B(t)$ can be mapped to a Kripke model $\mathcal{K}(t)$, whose accessibility relations are reflexive, symmetric and transitive. We then present a sound and complete translation to a classical planning problem which is quadratic to the number of possible initial states and is based on similar translations for conformant and contingent problems to classical ones.

In the second chapter of Part II, we extend the previous belief representation to accommodate on-line, multi-agent planning problems, where the plan must achieve the goal for one possible initial state: the true, hidden state. We introduce the notion of a planning agent and define how to evaluate truth in relation to the states that the planning agent considers possible at each given point in time. Our approach is based on a plan-execute-observe-and-replan cycle, for which we provide an algorithm and a new translation to classical planning. The new translation expresses a relaxation where the planning agent chooses one of the possible initial states as the assumed, hidden state (instead of being provided one from an external source). This assumed, hidden state is the source of all observations during planning. If the execution of the plan fails due to some observation which contradicts the assumption, then the assumed, true state is no longer considered as a possible candidate for the true, hidden state, and we replan. The translation is sound and complete, and the number of calls to the classical planner is bounded by the number of states and agents in the problem. We show that our approach can be used within the framework of generating dialogues. We present dialogues where agents can volunteer and ask for information, by considering what the other agents need to know or might know, in a goal-directed manner.

In Part III we provide two optimizations: a translation which is linear in the number of possible initial states and a decomposition approach. Though none of the two optimization is general enough to be applied to the entirety of our problems, we show that at least one optimization can be applied to most of the problems that are presented in this dissertation. For a linear translation to be possible, passive sensors must contain only static literals, while sensing actions must either contain only static literals or involve all agents (e.g. public announcements). A preprocessing phase is necessary in order to determine which sensing actions and passive sensors allow an agent to distinguish between two states. The decomposition approach is based on dividing the original problem into subproblems. We identify subproblems by first partitioning the set of literals based on their initial relevance. Two literals are initially relevant if, by knowing the truth value of one, an agent can derive the truth value of the other. The second step is to define whether two literals are dynamically relevant due to some conditional effect of a physical action, a sensing action or passive sensor. For example, given the sensing action of a formula, two literals are dynamically relevant if they both appear in the formula. By determining the sets of relevant literals, we can define the decomposition of the original problem into subproblems. where each subproblem corresponds to a partial joint belief $B_i(t)$ whose states are partial states. We then provide a translation which is sound, complete, and quadratic to the number of partial states of the largest subproblem.

The formulations and experimental results presented in this dissertation have been published in the following articles:

- Filippos Kominis and Hector Geffner. *Beliefs in multiagent planning: From one agent to many*. In the 24th International Conference of Automated Planning and Scheduling (ICAPS-14), *Workshop on Distributed and Multi-Agent Planning*, pages 62-68. [Chapter 4]
- Filippos Kominis and Hector Geffner. *Beliefs In Multiagent Planning: From One Agent to Many*. In the 25th International Conference of Automated Planning and Scheduling (ICAPS-15). [Chapter 4]
- Filippos Kominis and Hector Geffner. *Multiagent Online Planning with Nested Beliefs and Dialogue.* In the 27th International Conference of Automated Planning and Scheduling (ICAPS-17). [Chapter 5]

Contents

Abstract VII

Resumen VII

Preface IX

List of Figures XVII

List of Tables XIX

I Background 1

1. Classical Planning 3
1.1. The Classical Planning Problem 3
1.2. Syntax and Semantics 4
1.3. Example 5
1.4. The Planning Domain Definition Language 7
1.4.1. Conditional Effects 7
1.4.2. Axioms 8
1.5. Complexity 10
1.6. Classical Planners 12

2. Planning with Incomplete Information 13
2.1. Introduction 13
2.2. Conformant Planning 14
2.3. Contingent Planning 17
2.4. Complexity 19
2.5. Translations 19

3. Multi-agent Planning and Epistemic Logic 23

3.1. Multi-agent Planning 23
3.1.1. Introduction 23
3.1.2. Taxonomy of Multi-agent Problems 24
3.2. Epistemic Logic 25
3.2.1. Introduction 25
3.2.2. Muddy Children Puzzle 26
3.2.3. Kripke Models and the Logical System S5 28
3.2.4. Dynamic Epistemic Logic 31
3.2.5. Complexity 35
3.2.6. Knowledge vs Belief 36
3.2.7. Discussion 36

II Belief Representations And Translations **39**

4. Linear Multi-agent Planning **41**
4.1. Introduction 41
4.2. Language 43
4.3. Dynamics of Knowledge Updates 43
4.4. From $B(t)$ to Kripke Structures 46
4.5. Examples 47
4.5.1. Selective Communication 47
4.5.2. Collaboration through Communication 48
4.6. Translation into Classical Planning 49
4.7. Experiments 51
4.7.1. Active Muddy Child 53
4.7.2. Sum 53
4.7.3. Word Room 54
4.8. Relation to Single Agent Beliefs and DEL 55
4.9. Conclusion 56

5. Online Planning and Dialogues **57**
5.1. Introduction 57
5.2. Motivation 58
5.3. Language 59
5.4. Beliefs 60
5.4.1. External View 60
5.4.2. From Beliefs to Kripke Structures 61
5.4.3. Agent's View 61
5.5. Planning 62
5.5.1. Properties 64

5.5.2. Translation into Classical Planning . . . 65
5.5.3. Protocols . . . 66
5.6. From plans to dialogues . . . 68
5.7. Examples and experimental results . . . 69
5.7.1. Meeting problem . . . 69
5.7.2. Situated dialogue . . . 71
5.7.3. The Lights problem . . . 73
5.8. Related Work . . . 74
5.9. Conclusion . . . 75

III Optimizations and Variations **77**

6. A Linear Translation **79**
6.1. Requirements for a Linear Translation . . . 79
6.2. Preprocessing . . . 83
6.3. Linear Translation into Classical Planning . . . 84

7. A Decomposition Approach **87**
7.1. Decomposition through Relevance . . . 88
7.1.1. Initial Decomposition . . . 88
7.1.2. Dynamic Decomposition . . . 90
7.2. Belief Representation and Updates . . . 91
7.2.1. The Decomposed Joint Belief . . . 91
7.2.2. Updates . . . 91
7.3. From $B_D(t)$ to Kripke Structures . . . 92
7.4. Agent's view . . . 93
7.5. Example . . . 94
7.5.1. Decomposition . . . 94
7.5.2. Belief Representation . . . 96
7.6. Translation into Classical Planning . . . 98

8. Examples and Experiments **103**
8.1. Gossiping Problem . . . 103
8.2. Experiments . . . 105

IV Conclusions **109**
8.3. Contributions . . . 111
8.4. Ongoing and Future Work . . . 112
8.4.1. A Syntactic Approach . . . 112

8.4.2. Incorporating Belief 114

A. Properties of $K(P)$ translation **117**

B. Properties of $K(P, B(t), S_i(t))$ translation **123**

C. Properties of $K(P, B(t), S_i(t), O^+, O(t))$ translation **131**

D. Properties of $K(P, B_D(t), S_i^D(t))$ translation **139**

Bibliography **141**

List of Figures

1.1. A single agent planning problem, where the agent needs to pickup block B and move it to position I_g. 5
1.2. PDDL encoding of an action without conditional effects, an action with conditional effects and an axiom. 11

2.1. A conformant planning problem, where the agent needs to reach position I_g while having uncertainty about his initial position. I_1 to I_4 denote the possible initial positions of the agent. 16
2.2. A contingent plan for the problem in Example 2.3.1. 20

3.1. The Kripke model $\mathcal{K}$ of the Muddy Children problem *before* the father's announcement. 32
3.2. The Kripke model of the Muddy Children problem *after* the father's announcement. 32
3.3. Examples of action events for the Muddy Children problem. 34

5.1. Situated dialog example: on the left we see what agent a knows, in the middle what b knows and on the right the hidden, true state. Covered positions on the table indicate the positions that the respective agent cannot see. 71

7.1. The Kripke stuctures that correspond to $B_D(0)$ for the *Collaboration-through-communication* example. From left to right, and top to bottom, the Kripke structures correspond to $B_1(0)$, $B_2(0)$ and $B_3(0)$. Reflexive relations and agents' positions are omitted. 97

List of Tables

3.1. Epistemic axioms 30

4.1. Experimental results. Problems P shown on the left. The columns indicate number of atoms, actions, and axioms in $K(P)$, the number of possible initial states for P, and the resulting times and plan lengths. FF-X refers to the version of FF that supports axioms. The other columns refer to three different configurations of Fast Downward using the same search algorithm A* and the heuristics h_{max}, h_{cea} and h_{add}. The first configuration yields provably shortest plans. In the FF-X column, X/Y stands for X seconds and plan length Y. For Fast Downward, X-Y/Z stands for X seconds of total time, Y seconds spent on the search, and plan length Z. Unsolved problems indicated as "–". 52

8.1. Experimental results for the Public Gossiping problem with the decomposition approach, using A^* with the additive and the max heuristic. Times are in seconds. 104

8.2. Planning problems for which the linear translations and the decomposition approach can be used. 105

8.3. For the Collaboration through Communication, the number in the parenthesis is the number of blocks in the problem. For the Situated Dialogue, (X,Y) stands for X blocks whose position is unknown, with Y possible positions each. For the Public Gossiping, the number in the parenthesis is the number of secrets. In column #States we see the number of possible initial states of the original problem. In the decomposition column $l_1 - l_2 - ... - l_n$ stands for n partial beliefs, where l_j is the number of possible initial partial states of the partial belief $B_j(t)$ 106

8.4. Experimental results of optimizations in comparison with the quadratic translation. Times are in seconds and an X denotes that the approach was not applicable in the problem. 107

PART I

Background

CHAPTER 1

Classical Planning

1.1. The Classical Planning Problem

Classical planning is the task of finding a sequence of *deterministic* actions with *known* effects such that, when applied in the initial, *fully-known* state, it results in a state where the goal is satisfied. It can be formulated as a path-finding problem over a directed graph: nodes represent the different states of the environment, while edges are actions indicating the transition from one state to another due to the action. A *plan* then is a path from the node representing the initial state to a node where the goal is satisfied.

Definition 1.1.1. The classical planning model S is defined as the tuple $\langle S, s_0, S_G, A, f, c\rangle$ where:

- S is a finite and discrete set of states.
- $s_0 \in S$ is the known initial state,
- $S_G \subseteq S$ is a non-empty set of goal states
- $A(s) \subseteq A$ the set of actions in A that are applicable in each state $s \in S$,
- $s' = f(a, s)$ is a deterministic transition function which, given a state $s \in S$ and an action $a \in A(s)$, returns the resulting state s'.
- $c(a, s)$ is the positive cost of applying action a in the state s.

The solution to a classical planning model is called a *plan* π, where:

Definition 1.1.2. A plan π for a classical planning model S is a sequence of actions $\pi = [a_0, a_1, ..., a_n]$ such that, when applied to the initial state s_0, it results in a sequence of states $[s_0, s_1, ..., s_n]$ where $s_n \in S_G$.

1.2. Syntax and Semantics

In most cases, the set of possible states is too large to be represented explicitly. For this reason, factored representations are used. The most common representation is STRIPS (Fikes and Nilsson, 1971), which consists of only boolean variables (called fluents or facts or atoms) and allows states and actions to be described through these variables. Each boolean variable denotes whether a proposition about the world is true or false at a given state, and a state is defined as the set (implicitly a conjunction) of the truth values of all the variables in that state. The actions are defined based on their precondition and postconditions: the precondition of an actions is a set of fluents that need to hold in a state in order for the action to be applicable on that state, while post-conditions define which fluents become true/false after the action. Formally:

Definition 1.2.1. A classical planning problem P in STRIPS is defined as the tuple $\langle F, I, A, G \rangle$ where:

- F is the set of fluent symbols in the problem.
- I is the set of atoms over F which are true initially,
- A is the set of actions, where every $\alpha \in A$ is a tuple $\langle pre(\alpha), add(\alpha), del(\alpha) \rangle$, where $pre(\alpha), add(\alpha), del(\alpha) \subseteq F$,
- G a set of atoms over F that define the goal.

The set $add(\alpha)$ contains the atoms which are to become true when α is applied, while the set $del(\alpha)$ the atoms which are to become false. We assume that if a fact does not appear in I, then it is considered to be false (*closed-world assumption*).

A classical planning problem defines a classical planning model where:

- each state $s \in S$ is a subset of F, $s \subseteq F$, and if $p \in s$ then p is true, while for every $p' \in F$ s.t. $p' \notin s$, p' is false in s,
- the initial state s_0,

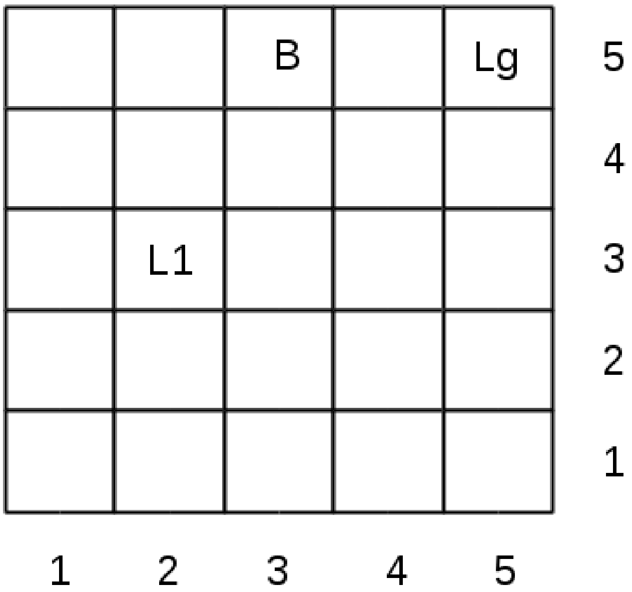

Figure 1.1: A single agent planning problem, where the agent needs to pickup block B and move it to position I_g.

- the goal states S_G are all the states $s \in S$ s.t. $G \subseteq s$,
- an action α is applicable in a state s, $\alpha \in A(s)$, iff $pre(\alpha) \subseteq s$,
- the transition function $f(\alpha, s)$ is defined by $add(\alpha)$ and $del(\alpha)$. More specifically, $f(\alpha, s)$ is $(s/\ del(\alpha)) \cup add(\alpha)$, given $\alpha \in A(s)$.

In the presence of a *cost function*, the evaluation of a plan is based on its total cost:

$$cost(\pi) = \sum_{j=0}^{|\pi|} cost(a_j, s_j)$$

If a cost function is not provided, we assume actions to have a uniform cost. i.e. each action has a cost 1. Based on the total cost of a plan, an optimal plan π^* is a plan with the minimum cost.

1.3. Example

Consider a single agent on a $N \times N$ grid, where $N = 5$. The agent is in position I_1 and the goal is for both the agent and the package B to be in position I_g, as depicted in Figure

1.1. We can model this problem as a classical planning problem $P = \langle F, I, O, G\rangle$ where:

- F : $\{a\_at\_x(p_i)$, $a\_at\_y(p_i)$, $p\_at\_x(p_i)$, $p\_at\_y(p_i)$, $holding\_package\}$, for $1 \leq i \leq N$, with $a\_at\_x(p_i), a\_at\_y(p_i)$ denoting the position of the agent and $p\_at\_x(p_i), p\_at\_y(p_i)$ denoting the position of the package.
- $I : \{a\_at\_x(p_2), a\_at\_y(p_3), p\_at\_x(p_3), p\_at\_y(p_5)\}$
- A :
  - action $\alpha = pick\_up(p_i, p_j)$, for any $1 \leq i < N$ and $1 \leq j < N$, with
    - $pre(\alpha)$: $a\_at\_x(p_i)$, $a\_at\_y(p_j)$, $p\_at\_x(p_i)$, $p\_at\_y(p_j)$
    - $add(\alpha)$: $holding\_package$
    - $del(\alpha)$: $p\_at\_x(p_i)$, $p\_at\_y(p_j)$
  - action $\alpha = drop(p_i, p_j)$, for any $1 \leq i < N$ and $1 \leq j < N$, with
    - $pre(\alpha)$: $a\_at\_x(p_i)$, $a\_at\_y(p_j)$, $holding\_package$
    - $add(\alpha)$: $p\_at\_x(p_i)$, $p\_at\_y(p_j)$
    - $del(\alpha)$: $holding\_package$
  - action $\alpha = up(p_i)$, for any $1 \leq i < N$, with
    - $pre(\alpha)$: $a\_at\_y(p_i)$
    - $add(\alpha)$: $a\_at\_y(p_{i+1})$
    - $del(\alpha)$: $a\_at\_y(p_i)$
  - action $\alpha = down(p_i)$, for any $1 < i \leq N$, with
    - $pre(\alpha)$: $a\_at\_y(p_i)$
    - $add(\alpha)$: $a\_at\_y(p_{i-1})$
    - $del(\alpha)$: $a\_at\_y(p_i)$
  - action $\alpha = left(p_i)$, for any $1 < i \leq N$, with
    - $pre(\alpha)$: $a\_at\_x(p_i)$
    - $add(\alpha)$: $a\_at\_x(p_{i-1})$
    - $del(\alpha)$: $a\_at\_x(p_i)$
  - action $\alpha = right(p_i)$, for any $1 \leq i < N$, with

- $pre(\alpha) : a\_at\_x(p_i)$
- $add(\alpha) : a\_at\_x(p_{i+1})$
- $del(\alpha) : a\_at\_x(p_i)$

- $G : a\_at\_x(p_5), a\_at\_y(p_5), p\_at\_x(p_5), p\_at\_y(p_5),$

One of the plans for this problem is:

$$\pi = \{up(p_3), up(p_4), right(p_2), pick\_up(p_3, p_5),\\ right(p_3), right(p_4), drop(p_5, p_5)\} \quad (1.1)$$

1.4. The Planning Domain Definition Language

The Planning Domain Description Language (PDDL) (McDermott, 2000) has become the standard language for expressing classical planning problems. PDDL allows the usage of STRIPS, as well as extensions of it, and different versions of PDDL denote differences in the expressivity of the language (Fox and Long, 2003; McDermott, 2003). In this dissertation, we use two specific extensions of STRIPS: *conditional effects* and *axioms*.

1.4.1. Conditional Effects

Conditional effects refer to effects that are conditioned on the truth values of fluents: if the condition holds the effect takes place, otherwise it is not.

Definition 1.4.1. Given a classical planning problem P with a set of actions A, each action $\alpha \in A$ is defined as a tuple $\langle pre(\alpha), \mathit{eff}(\alpha)\rangle$ where:

- $pre(\alpha) \subseteq F$ of P,
- $\mathit{eff}(\alpha) = \{e_1, ..e_n\}$, where $e_j = c_j \rightarrow add_j, del_j$

and c_j, add_j, del_j are sets of fluents. The set c_j is the condition of the effect, while add_j and del_j represent its effects. In addition, c_j can be the expression $true$, indicating that the effect can take place in all states where the action can be applied.

As before, $\alpha \in A(s)$ if $pre(\alpha) \subseteq s$, and $f(\alpha, s)$ stands for $(s/\ Dels(\alpha, s)) \cup Adds(\alpha, s)$, where:

$$Dels(\alpha, s) = \bigcup_{c_j \subseteq s} del_j$$
$$Adds(\alpha, s) = \bigcup_{c_j \subseteq s} add_j$$

for all $e_j \in \mathit{eff}(\alpha)$.

The applicability of an action in a state depends on its preconditions, and the applicability of each of its (conditional) effects on their respective conditions.

1.4.2. Axioms

Actions define a relationship between two states, as we saw with the definition of the transition function $s' = f(\alpha, s)$: the truth value of fluents in state s' depend on the truth value of fluents in state s and the action α. An axiom, on the other hand, defines a relationship between the truth values of fluents and derived predicates in the *same* state: whether a derived predicate is true depends on whether some other fluent(s) is true in the same state (McDermott, 2000).

We define *derived atoms* as the atoms whose truth value depends only on the truth values of (i) other derived atoms, and/or (ii) non-derived, called *primitives*, fluents. This means that a derived atom cannot be explicitly defined as true in the initial state - only through explicitly defining the truth value of the primitives it depends on - and its truth value cannot be changed directly through some action's effects. On the other hand, they can be used in preconditions, in the conditions of conditional effects and in the goal. The presence of derived predicates allows us to extend the definition of a classical planning problem:

Definition 1.4.2. A classical planning problem with axioms is the tuple $P = \langle F, I, A, X, G \rangle$ where:

- $F = F_d \cup F_p$, where F_d is the set of derived atoms and F_p the set of primitive ones, with $F_d \cap F_p = \emptyset$
- I is the set of literals over F_p which are true initially,
- A is the set of actions, where every $\alpha \in A$ is a tuple $\langle pre(\alpha), \mathit{eff}(\alpha) \rangle$, where $pre(\alpha) \subseteq F$ and for every $e_i \in \mathit{eff}(\alpha)$ we have that $c_i \subseteq F$ and $add_i, del_i \subseteq F_p$.

- G a set of atoms over F that define the goal.
- X is the set of axioms, where every $a_x \in X$ is a pair $\langle p_x, \mathcal{F}_x \rangle$, where $p_x \in F_d$ and $\mathcal{F}_x$ a formula in first order logic.

Two clarifications are necessary: first, given an axiom α_x, p_x can only be positive. That is, we cannot derive the negation of a derived atom, yet there is no need to because atoms that do not explicitly appear true in a state are assumed to be false (closed world assumption). Secondly, in order to not get trapped in a circular definition and, at the same time, be able to use derived fluents in the first-order formula $\mathcal{F}_x$ of another derived fluent, the set of axioms X must be stratified (Thiébaux et al., 2005).

Definition 1.4.3. An axiom set X is stratified iff there exists a partition of the set of derived fluents F_d into non-empty sets $\{F_d^i, 1 \leq i \leq n\}$ such that for every $p_i \in F_d^i$ and every axiom $\alpha_x = \langle p_i, \mathcal{F}_x \rangle \in X$:

1. if $p_j \in F_d^j$ appears in NNF($\mathcal{F}_x$), then $j \leq i$,
2. if $p_j \in F_d^j$ appears negated in NNF($\mathcal{F}_x$), then $j < i$

Any stratification of F_d induces a stratification on X. Applying all axioms in each stratum X^i before applying the axioms in stratum X^{i+1}, always leads to the same fixed point independently of the chosen stratification.

Though axioms are not strictly necessary, since they can be compiled away, they prove to be useful, both in terms of modeling PDDL domains, as well as computational efficiency. First, they provide an intuitive, natural way of defining recursive relationships, such as reachability. There are no intuitive ways to define such relationships without axioms in PDDL, and such ways usually lead to complex action definitions. Second, compiling axioms away leads to either domain descriptions which, in the worst-case, are exponential in size to the original one, or to plans of worst-case exponential length (Thiébaux et al., 2005).

Example with conditional effects and axioms

In Figure 1.2 we can see part of the PDDL encoding: action $up(p_3)$ and action up with conditional effects, taken from the problem we described in Example 1.3. We also show the PDDL encoding of an axiom: the derived atom "at-corner" represents the fact that the agent is present in one of the corners of the grid, while the formula defining its truth value is written in disjunctive normal form (DNF) where every term denotes one of the corner positions.

- action $\alpha = pick\_up$, with
  - $pre(\alpha)$ is empty.
  - $\mathit{eff}(\alpha) = a\_at\_x(p_i)$,$a\_at\_y(p_j)$,$p\_at\_x(p_i)$,$p\_at\_y(p_j) \rightarrow$ $holding\_package$,$\neg p\_at\_x(p_i)$,$\neg p\_at\_y(p_j)$, for $1 \leq i < N$ and $1 \leq j < N$.
- action $\alpha = drop$, with
  - $pre(\alpha) = holding\_package$
  - $\mathit{eff}(\alpha) = a\_at\_x(p_i)$,$a\_at\_y(p_j)$, $\rightarrow$ $p\_at\_x(p_i)$,$p\_at\_y(p_j)$,$holding\_package$, for $1 \leq i < N$ and $1 \leq j < N$.
- action $\alpha = up$, with
  - $pre(\alpha)$ is empty.
  - $\mathit{eff}(\alpha) = at\_y(p_i) \rightarrow at\_y(p_{i+1}), \neg at\_y(p_i)$, for $1 \leq i < N$.
- action $\alpha = down$, with
  - $pre(\alpha)$ is empty.
  - $\mathit{eff}(\alpha) = at\_y(p_i) \rightarrow at\_y(p_{i-1}), \neg at\_y(p_i)$, for $1 < i \leq N$.
- action $\alpha = left$, with
  - $pre(\alpha)$ is empty.
  - $\mathit{eff}(\alpha) = at\_x(p_i) \rightarrow at\_x(p_{i-1}), \neg at\_x(p_i)$, for $1 < i \leq N$.
- action $\alpha = right$, with
  - $pre(\alpha)$ is empty.
  - $\mathit{eff}(\alpha) = at\_x(p_i) \rightarrow at\_x(p_{i+1}), \neg at\_x(p_i)$, for $1 \leq i < N$.

in which case, one plan could be:

$$\pi = \{up, up, right, right, right\}$$

1.5. Complexity

Given a classical problem P in STRIPS, the problem of determining whether a plan exists for an arbitrary problem instance is PSPACE-complete (Bylander, 1994). This is

```
(:action up-p3
      :precondition  (at-y  p3)
      :effect  (and  (at-y  p4)  (not  (at-y  p3)))
)


(:action up
      :effect  (and
                 (when  (at-y  p1)  (and  (at-y  p2)  (not (at-y  p1))))
                 (when  (at-y  p2)  (and  (at-y  p3)  (not (at-y  p2))))
                 ...
                 (when  (at-y  p4)  (and  (at-y  p5)  (not (at-y  p4))))
                )
)

(:derived (at-corner)
     (or
     (and  (at-x  p1)  (at-y  p1))
     (and  (at-x  p5)  (at-y  p5))
     (and  (at-x  p1)  (at-y  p5))
     (and  (at-x  p5)  (at-y  p1))
     )
)
```

Figure 1.2: PDDL encoding of an action without conditional effects, an action with conditional effects and an axiom.

the class of problems that can be solved using memory which is polynomial to the input size and unrestricted amount of time. The problem of finding optimal plans (minimum cost) for arbitrary instances also belongs in the same class. Given that in the worst case, planning problems are intractable, current approaches are generally assessed in terms of performance (memory/time) and/or plan quality on benchmarks. There are no complexity results for dynamic epistemic logic with common knowledge.

1.6. Classical Planners

Despite the theoretical worst-case results, classical planning approaches have proved to do quite well in terms of performance, evident by the results of the International Planning Competitions (IPC). The situation is similar to satisfiability problems: while difficult in the worst case to tackle, current approaches do well in terms of time and problem-size's they tackle (Kautz and Selman, 1996; Gomes et al., 2008).

There are two techniques that allow classical planners to do so well. The first is heuristic search, with the first heuristic planner being HSP (Bonet and Geffner, 1999). Since then, the planners that have dominated the IPC use heuristic search (Bonet and Geffner, 2001; Hoffmann and Nebel, 2001; Helmert, 2006; Richter and Westphal, 2010), or portfolios techniques which combine different heuristics and search algorithms (Vallati et al., 2015). Heuristics estimate the distance of a state to the goal, and, based on that distance, states can be pruned (and thus reduce the search space) or have less priority concerning expansion.The second technique is search enhancements, like helpful actions (Hoffmann and Nebel, 2001; Helmert, 2006) and landmarks (Richter and Westphal, 2010). Helpful actions are used to reduce the branching factor of the search. Just like heuristics can be used to reduce the number of states to be considered, helpful actions reduce the number of the applicable actions that need to be checked. Landmarks are propositional formulas than have to become true at some point in all plans. They can be considered as subgoals of the original problem and used for measuring the distance of a state to a goal state, based on the number of unachieved landmarks.

Classical planning is more than heuristics guiding the search for plans. It tries to (i) explain how certain heuristics are related to each other (Helmert and Domshlak, 2009), and (ii) understand the nature of the heuristics, by finding connections with different approaches (Bonet and Geffner, 2008). Recently, it has moved from taking advantage of heuristics to taking advantage of the *structure* of the planning problems, as captured by the notion of *width* (Lipovetzky and Geffner, 2012; Lipovetzky and Geffner, 2017).

CHAPTER 2

Planning with Incomplete Information

2.1. Introduction

In the previous chapter we considered the model of classical planning that assumes deterministic actions and a fully known initial state. These two assumptions are quite strong, especially if we consider real-life problems where (i) having complete information is either too difficult or impossible, and (ii) predicting the effects an action is not always feasible. Such scenarios include agents being placed in unknown environments, like the ruins of a building after a natural disaster, with actions in their disposal which might fail, like picking up a rock, in which case the agent might need to retry until he succeeds.

The notions of uncertainty and partial observability are important for this dissertation when multiple agents are acting in the same environment. The agents may originally know different aspects of the environment (like different parts of the same map), they might sense different facts about the environment while acting in it and they can communicate in order to reduce their uncertainty.

In single-agent scenarios, uncertainty is limited to the environment: there are facts whose truth value is not known to the agent. In multi-agent scenarios, uncertainty is about the environment but also about the knowledge that other agents have, or if we take it one step further: the knowledge other agents have concerning the knowledge other agents have, and so on.

In this chapter, uncertainty denotes the fact that the initial state is not known to the agent and partial observability denotes the agent's ability to observe certain facts about the world based on some condition, and we will refer to it as *sensing*. Also, we will assume that all actions are deterministic.

2.2. Conformant Planning

The deterministic conformant planning model is the classical planning model where the initial situation is not fully known. This means that there is no specific initial state, but a set of possible initial states. The conformant planning problem is to find a sequence of actions such that its execution guarantees that the goal is achieved independently of the actual true state.

A conformant planning problem can be formulated as a path-finding problem over a directed graph, just like the classical planning one. The key difference is that nodes are not states as defined in classical planning, but *belief states*: sets of states the agent considers possible in a given situation (Bonet and Geffner, 2000). It is this distinction that points out the role of actions with conditional effects: an action may be executable in the belief state but have no effects, or, even more interestingly, each state in the belief state might be affected differently by the applied action, depending on which conditional effects take place.

Definition 2.2.1. A deterministic conformant planning model Q is a tuple $\langle S, S_0, S_G, A, f\rangle$ where:

- S is a finite set of states,
- $S_0 \subseteq S$ is the set of initial states,
- $S_G \subseteq S$ is the set of goal states,
- A is a set of actions, with $A(s)$ denoting the set of actions applicable in state $s \in S$,
- a transition function $s' = f(\alpha, s)$, for $\alpha \in A(s)$.

We now define the representation of a deterministic conformant problem.

Definition 2.2.2. A deterministic conformant planning problem P is a tuple $\langle F, I, O, G\rangle$, where:

- F stands for the fluents or atoms of the problem,
- I is the set of clauses over F that define the set of initial states,
- A the set of deterministic actions,
- G a set of literals over F defining the goal.

The deterministic conformant planning model P is the same as the classical planning one, with the difference that the initial state s_0 is replaced by S_0, the initial set of states. That is, the deterministic conformant problem defines a model where the states s of the state space S are set of literals that represents truth values over all literals in F, the initial belief state b_0 is composed of all the states s that satisfy all clauses in I, the set of goal states is composed of all the states s where $G \subseteq s$, an action α belongs to $O(s)$ if $\alpha \in O$ and $pre(\alpha) \subseteq s$, and the deterministic state transition function $s' = f(\alpha, s)$, which maps pairs of action-state to states.

By extension, since an action is applicable on a state if its preconditions hold in the state, an action is applicable in a belief state if the preconditions of the actions hold in all the states in the belief state. Formally, if $A(b)$ is the set of all actions applicable in the belief state b, we define:

$$A(b) = \{\ \alpha \mid f(\alpha, s) \neq \emptyset, s \in b, \alpha \in A\}$$

Similarly, applying an action α to a belief state b, at time t results in

$$b_{t+1} = \{s' \mid s' = f(\alpha, s), s \in b_t\}$$

We write $P|_s$ to refer to the classical planning problem P which is like the conformant planning problem except the initial belief state contains only the state s.

Definition 2.2.3. A conformant plan π for a deterministic conformant planning problem P is a sequence of actions $\alpha_0, \alpha_1, ..., \alpha_n$ such that π is a solution to the classical planning problem $P|_s$, for each possible initial state s of S_0.

Example 2.2.1. *Consider a single agent on a $N \times N$ grid. I_g indicates the goal position of the agent, and initially the agent has uncertainty about his position. I_1 to I_4 indicate the possible initial positions of the agent.*

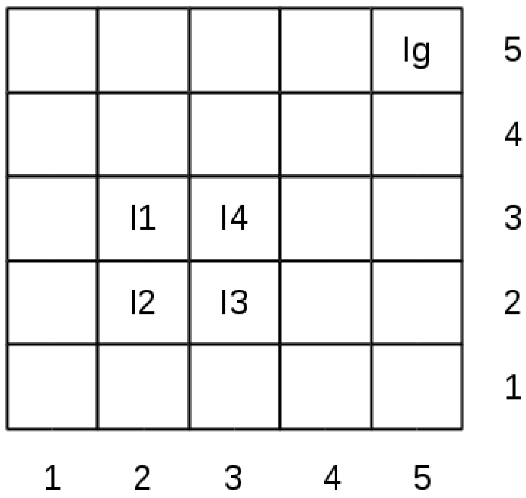

Figure 2.1: A conformant planning problem, where the agent needs to reach position I_g while having uncertainty about his initial position. I_1 to I_4 denote the possible initial positions of the agent.

This example can be formulated as a deterministic conformant problem $P = \langle F, I, O, G \rangle$, with:

- $F : at\_x(p_i), at\_y(p_i)$, for $1 \leq i \leq 5$
- $I : oneof(at\_x(p_2), at\_x(p_3)), oneof(at\_y(p_2), at\_y(p_3)), \neg at\_x(p_j), \neg at\_y(p_j)$, for $j \in \{1, 4, 5\}$.
- O is the set of actions $up, down, left, right$ as defined in the example of section 1.3.
- $G = at\_x(p_5), at\_y(p_5)$.

The $oneof(l_1, .., l_n)$ expression stands for the fact that only one l_j can be true. In this example, the $oneof$s represent that the agent can be in one out of two specific rows and one out of two specific columns.

A plan for this conformant problem is:

$$\pi = \{up, up, up, right, right, right\}$$

which guarantees that the goal will be achieved, no matter which of the four possible initial states is the true one. This is achieved by the third repetition of the actions up and $right$. After the execution of the first three actions, the agent is at the top row of the grid with certainty, either at (2,5) or at (3,5). Executing the rest of the plan results to either the agent achieving the goal with the last action, or achieving the goal with the previous to last action - in which case the execution of the last action has no effect (since no conditional effect take place) and the agent remains in the same position.

2.3. Contingent Planning

Contingent planning extends conformant planning with sensing actions. It deals with problems were the agent has some uncertainty about the environment initially, as in conformant planning problems, but he is able to receive partial information about the true state through his sensors. Planning with uncertainty and partial observability of the environment is viewed a complex form of planning: both classical and conformant planning are special cases of contingent planning.

The contingent planning model extends the conformant one with a deterministic sensor model (Geffner and Bonet, 2013).

Definition 2.3.1. The deterministic contingent planning model is the tuple $Q = \langle S, S_0, S_G, A, f, O, n\rangle$ where:

- S is a finite set of states,
- $S_0 \subseteq S$, the initial belief state,
- $S_G \subseteq S$, the set of goal states,
- A is a set of actions,
- a transition function $s' = f(\alpha, s)$, for $\alpha \in A(s)$.
- O is the set of observations tokens,
- n is a sensor model $O(s, a) \in O$, that associates state-actions pairs to a single observation token.

In terms of belief states, $b' = f(b, \alpha)$ is the belief state b' resulting after applying action α to the belief state b. The belief state b^o is the belief state resulting after an observation o on a belief state b and it contains only the states $s \in b$ that are compatible with the observation.

Definition 2.3.2. A contingent planning problem is a tuple $P = \langle F, I, A, O, G\rangle$, where:

- F a set of Boolean fluents
- I a set of clauses over F defining the initial situation,
- A a set of actions, where each action $\alpha \in A$ is a tuple $\langle pre(\alpha), \mathit{eff}(\alpha)\rangle$, and $\mathit{eff}(\alpha)$ is a set of conditional effects e, where $e_i : c_i \rightarrow add_i, del_i$, each a conjunction of literals in F,
- O a sensor model, where each $o \in O$ is a tuple $\langle C, l\rangle$, where C is a set of literals and l a positive literal in F.
- $G \subseteq F$, describing the set of goal states.

A tuple $o = \langle C, l\rangle$ indicates that the truth value of l is observable to the agent if C holds in the state.

While a plan for classical and conformant planning problems is a sequence of actions, in the case of contingent planning problems the plan is a tree whose nodes are mapped to actions. If the action results in an observation then the node has two children representing the two possible outcomes of the sensing. In other words, the branching represents the two different observations.

Example 2.3.1. *We extend the example of 2.2.1 to a contingent planning problem with sensing. Specifically, the agent can observe whether he is next to a wall.*

We model the problem as a deterministic contingent planning problem $P = \langle F, I, A, O, G\rangle$ where:

- $F : w_u, w_l, w_d, w_r, at\_x(p_i), at\_y(p_i)$, for $1 \leq i \leq 5$
- $I\ :\ oneof(at\_x(p_2), at\_x(p_3)), oneof(at\_y(p_2), at\_y(p_3)), \neg at\_x(p_j), \neg at\_y(p_j)$, for $j \in \{1, 4, 5\}$.
- A is the same as in the classical planning version of the same example.
- O :
  - sense-upper-wall: $\langle\{at\_y(p_5)\}\ ,\ w_u\rangle$
  - sense-left-wall: $\langle\{at\_x(p_1)\}\ ,\ w_l\rangle$
  - sense-right-wall: $\langle\{at\_x(p_5)\}\ ,\ w_r\rangle$
  - sense-down-wall: $\langle\{at\_y(p_1)\}\ ,\ w_d\rangle$
- $G = at\_x(p_5), at\_y(p_5)$.

A contingent plan for this problem can be seen in Figure 2.2. The first observation appears after the second action up since the outcome of all sensing actions before that can be predicted in terms of the uncertainty the agent has initially.

2.4. Complexity

The problem of showing whether there is a valid plan for an arbitrary deterministic conformant problem instance is PSPACE-complete and for non-deterministic conformant problems EXPSPACE-complete (Haslum and Jonsson, 1999). The problem of showing whether a valid plan for an arbitrary deterministic contingent problem instance exists is shown to be EXPSPACE-complete (Rintanen, 2004).

Classical planning appears to be the easiest form of planning when we consider plan verification: determining whether there is a classical plan with length at most k (where k is polynomial in the size of the problem) is NP-complete, while the same problem for conformant planning is Σ_2^P-complete (Turner, 2002).

2.5. Translations

Conformant and contingent planning must address two main issues: the first is the problem of belief tracking and how to represent those beliefs in a compact way. The second problem is having informed heuristics over those beliefs for solving the problems. Translation approaches (Palacios and Geffner, 2006; Palacios and Geffner, 2007; Albore et al., 2009; Albore et al., 2010; Palacios and Geffner, 2009) address these issues by compiling the deterministic conformant problem to a classical planning problem, and the deterministic contingent problem to a fully-observable non-deterministic (FOND) one. Such translations allow the usage of heuristics and search enhancements implemented by classical and FOND planners.

These translations are of special interest for us, since our work is built on similar ideas. In conformant translations, like K_O (Palacios and Geffner, 2009), belief states are represented as a plain state. This is achieved by using an epistemic encoding which maps the belief state to a set of epistemic literals, representing the uncertainty of the agent explicitly through literals KL, denoting that L is known to the agent. Actions then make changes on the knowledge level (Petrick and Bacchus, 2002). An extension of K_0 is the translation $K_{T,M}$, where T is a set of tags and M a set of merges. The idea behind $K_{T,M}$ is that, through T, they can track truth values of literals in a state through tracking what

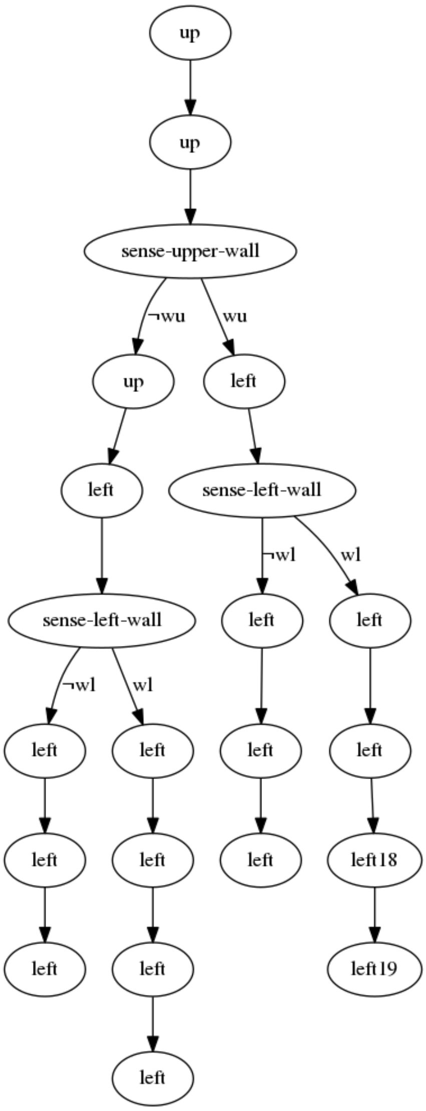

Figure 2.2: A contingent plan for the problem in Example 2.3.1.

was true initially in that state. Given a literal L and a tag $t \in T$, KL/t stands for *"if t is true initially, then L is true"*. Merges $m \in M$ are pairs of a set of tags in T and a literal L and are used to define when KL. The intuition is that if L is known to hold over all tags in M, then L is known to hold in the current belief state.

We will appeal to these translations below.

CHAPTER 3

Multi-agent Planning and Epistemic Logic

3.1. Multi-agent Planning

3.1.1. Introduction

In the planning problems we have seen so far, a single agent is planning, acting and sensing. A multi-agent planning problem is the problem of planning in the presence of multiple agents who also plan and act in the same environment.

Several real-world applications are multi-agent planning problems. For example:

- RoboCup, where a team of agents is playing against another team in the game of football.
- Exploration problems, where a number of autonomous robots are mapping an area, e.g. drones over inhabited areas identifying fields or structures.
- Search and Rescue problems, where robots are deployed to areas affected by a disaster, looking for survivors or recording damages.
- Logistics problems, where a group of robots need to coordinate their plans in order to deliver products or working within an industrial environment, e.g. assembly lines.

All of the above are multi-agent problems with very different properties and planning approaches. In the following section we will identify a taxonomy of multi-agent problems based on these properties and mention some of the techniques currently used to solve them.

3.1.2. Taxonomy of Multi-agent Problems

Multi-agent planning cannot be defined with a single model, since there are families of multi-agent planning problems where many different factors are involved (Weiss, 1999). We will present some of these factors, relevant to our work.

A first taxonomy can be based on whether the agents can achieve the goal by themselves or they need to collaborate with other agents (Brafman and Domshlak, 2008)

- Tightly-coupled problems are the problems where agents must coordinate in order to achieve the goal.
- Loosely-coupled problems are the problems where the need for interaction among the agents is small.

An example for the first case would be a logistics problem, where in order for the package to be moved from one location to another, one agent must make sure that the package will be in the first location, and another agent will pick it up from the second. An example for the second case would be an exploration problem, where each agent is assigned an area to explore and his plan would not intervene with the plans of the other agents. On the other hand, if the agents exploring share finite resources, such as power stations for charging, then the sub-tasks of the problem are strongly related.

Multi-agent problems can be also categorized based on the relation between the agents (Shamma, 2008; Stone, 2002):

- Collaborative agents with a common goal/utility .
- Adversarial agents trying to optimize their own utility.

In a logistics problem agents must cooperate in order to achieve the goal, which is the delivery of a package (as long as the agents do not try to optimize their own utility function but a common one). On the other hand, security problems are examples of adversarial agents: an agent trying to prevent another agent from entering a facility undetected, or from gaining access to a system remotely.

Another distinction is based on communication (Goldman and Zilberstein, 2003; Brafman and Beer Sheva, 2015):

- Agents can share everything or agents much preserve some privacy.
- Communication is not possible, limited, or without restrictions.

In the logistics example, privacy might be important since some agents do not want to share e.g. the names of their clients or the final destinations of their packages etc. In the exploration problem, communication might be necessary if the agents need to share their current state (position, fuel needs, collaboration requests) but not possible (e.g. in a hostile environment) or limited (due to resources).

Last, there is an important distinction between planning *for* the agents or *by* the agents (De Weerdt and Clement, 2009):

- Centralized approach, where an agent plans for all agents.
- Decentralized approach, where all agents plan.

In the centralized approach, an agent must be aware of the capabilities (set of actions) of each agent and of their goals in order to plan for them.In the decentralized approach, each agent plans for himself and coordination of plans must be achieved either before (based on some convention) or during the planning phase (Jonsson and Rovatsos, 2011; Tonino et al., 2002).

Choosing the approach depends on the problem at hand. Centralized planning is useful when optimal (or close to optimal) plans are necessary or when communication is limited (in centralized planning there are two phases of communication: one for receiving information from the agents, and two, for sending the corresponding sub-plans to their agents). Decentralized planning is useful when we give priority to scalability, privacy and robustness. On the other hand, in strongly-related tasks, decentralized approaches depend on conventions being agreed upon before planning, or communication in order to achieve coordination.

In this dissertation we are interested in **collaborative**, multi-agent planning problems with partial observability, where the agents share a **common goal**, communication is *without restrictions*, and we use a **centralized** approach for solving them.

3.2. Epistemic Logic

3.2.1. Introduction

Planning agents make decisions that allow them to achieve their goal in a given environment. Their decisions are based on what they *know* about the environment and their actions. For example, in classical planning the agent knows everything about the world, while in contingent planning the agent knows only parts of it and uses sensing to get more information.

When a planning agent is placed in an environment with multiple agents who also act, more things need to be taken into account. For example, what do the other agents know or don't know, what do they know that I know, which information is common knowledge etc. Reasoning by taking into consideration the *knowledge* the other agents have, allows the agent to make more intelligent decisions. For example, he will not ask another agent about some information that he already knows the other agent cannot possess, and he will not share information with agents he knows are already aware of them. In other words, the planning agent must be able to reason by considering not only the state of the world but the *epistemic states* of the other agents.

Epistemic logic is the logic of knowledge and belief (Hintikka, 1962). It is a formal approach that allows the study of individual, group and common knowledge based on what information is available to each agent. With roots in modal logic (Van Benthem et al., 2010) and Kripke semantics (Kripke, 1963), it offers a formulation which allows us to evaluate, given a situation, propositions such as "Ann does not know the light is on", "Bob knows Ann does not know the light on" etc. Though epistemic logic is quite expressive in the problems it can model, in this thesis we will make use of a part of its expressiveness: we are interested in the notion of knowledge (and not of belief), the notion of multiple agents (of which the single-agent is a special case), the idea of *public* actions (in contrast with *private*) and the notion of common knowledge.

3.2.2. Muddy Children Puzzle

The Muddy Children puzzle (Fagin et al., 1995) is a representative example of how knowledge and lack of knowledge are communicated and the effect they can have. Here we present one version of the puzzle.

Example 3.2.1. *Three children, who are perfect logicians, are playing in the mud. The father calls them to the house and makes sure that each child can see every other child. He, then, tells them "At least one of you has mud on his forehead". He then says "If you know your forehead is muddy say it now". No child speaks. Father repeats "If you know your forehead is muddy, say it now". Some of the children say that they know. How many children have muddy foreheads?*

The above puzzle can be seen in different variations, where the context or the end question changes. In all cases, the Muddy Children puzzle is a classic example of:

- The role of common knowledge: the father's first announcement "At least one of you has mud on his forehead" makes common knowledge that fact - all children know that all children know that all children know.. that not all of them have clean foreheads.

- The role of communicating knowledge or lack of it: a child not speaking, which has the same effect as saying it does not know if he has mud on its forehead, provides information which is necessary for the rest of the children.

Since we will study the example in detail later on, we will give a general idea of the solution to the problem here. Suppose that the names of the children are Ann, Bob and Cadie.

Suppose the only one with a muddy forehead is Ann (which is not the case in the example). Ann sees Bob and Cadie clean, and the father says that at least one of them is muddy. Ann can derive that it must be her who has the muddy forehead. When the father asks them to say whether they know they have a muddy forehead, Ann will reply positively. From the point of view of Bob, Bob sees Ann with a muddy forehead but has no knowledge about his forehead. He also sees Cadie is not muddy, and he knows that Ann can also see Cadie. When Ann says that she knows she is muddy, Bob can infer that the only way for Ann to know that is if himself and Cadie are clean. Cadie can do the same inference. After Ann's announcement of knowing she has a muddy forehead, both Bob and Cadie will get to know they are not muddy.

Now, suppose both Ann and Bob have muddy foreheads. The first time the father asks, no one will reply since Ann and Bob see at least one more child muddy. From the point of view of Ann, Bob has a muddy forehead while Cadie does not. She can infer that if herself did not have a muddy forehead, it would mean that Bob sees both of them clean, which means that he would have replied after the father asked. Since Bob did not reply, it must be that he sees at least one child with a muddy forehead and Cadie is clean, so she infers it must be her. Thus, the second time the father asks, she replies that she does know she is muddy. Bob does the same for the same reason. It is easy to show that if m, out of a total number of n, children are muddy, the muddy children will reply that they know they are muddy the m^{th} time the father asks - which means that they will get to know if they are muddy after the $(m-1)^{th}$ question.

3.2.3. Kripke Models and the Logical System S5

Language

Given P, a finite set of atomic propositions, and A the finite set of agents (or agent-symbols) the language of (multi-agent) epistemic logic $\mathcal{L}$ with common knowledge (Van Ditmarsch et al., 2007) is generated by the following BNF:

$$\phi ::= p \mid \neg\phi \mid (\phi \wedge \phi) \mid K_i\phi \mid C\phi$$

where $p \in P$ and $i \in A$. The K_i is the knowledge operator, the C the common knowledge operator, and we will refer to $K_i\phi$ and $C\phi$ as epistemic literals, and formulas which which contain epistemic (and, possibly, propositional) literals as epistemic formulas. For every agent i, $K_i\phi$ will be read as *"i knows that ϕ is true"*, while $C\phi$ as *"ϕ is common knowledge"*. The sentence *"i knows the truth value of ϕ"* corresponds to $K_i\phi \vee K_i\neg\phi$, and *"i does not know the truth value of ϕ"* to $\neg K_i\phi \wedge \neg K_i\neg\phi$. Lastly, common knowledge can be defined over a set of agents, for example $C_B\phi$ where $B \subseteq A$, which is read as *"ϕ is common knowledge among all agents in B"*. Nevertheless, from here on we will assume that the common knowledge operator is over all agents ($B = A$).

Returning to the Muddy Children puzzle, we have $F = \{m_a, m_b, m_c\}$, where m_i indicates that i is muddy, and $A = \{a, b, c\}$, for Ann (a), Bob (b) and Cadie (c). After the father's announcement that at least one child is muddy, we have $C(m_a \vee m_b \vee m_c)$, while the formula that expresses that at least one child knows whether it's muddy would be:

$$(K_a m_a \vee K_a\neg m_a) \vee (K_b m_b \vee K_b\neg m_b) \vee (K_c m_c \vee K_c\neg m_c)$$

If we wanted to write that all children that are muddy know that they are muddy:

$$(m_a \to K_a m_a) \wedge (m_b \to K_b m_b) \wedge (m_c \to K_c m_c)$$

and the fact that Bob knows that Ann knows whether Cadie is muddy, as:

$$K_b(K_a m_c \vee K_a\neg m_c)$$

Semantics

The semantics of epistemic logic are based on Kripke models (Kripke, 1963). Informally, a Kripke model is a representation of states and indistinguishability between states. States correspond to truth-valuations over P, while the indistinguishability between two states for an agent, is denoted by the existence of the accessibility relations, which represent the fact that that agent cannot distinguish between the two states - from the point of view of the agent, they are equally possible.

Definition 3.2.1. A Kripke model is a structure $\mathcal{K} = \langle W, R, V \rangle$, where

- W is a set of state names.
- R is a set of accessibility relations $R_i \subseteq W \times W$, for every $i \in A$. We will write $R_i(s, s')$ to denote that agent i cannot distinguish between states s and s'.
- $V : P \to 2^S$ is a valuation function that for every $p \in P$ returns the set of states where p is true. We will also write $V(s)$ to denote the set of atoms that are true in state s.

A pointed Kripke model is a pair $(\mathcal{K}, s)$, where $\mathcal{K}$ is a Kripke model and s a state s.t. $s \in W$.

Given Definition 4.3.1, we define how to interpret formulas on some state s of a given Kripke model $\mathcal{K}$.

Definition 3.2.2. We define that a formula ϕ is true in a state s of a given Kripke model $\mathcal{K}$, also written as $\mathcal{K}, s \models \phi$, as:

- $\mathcal{K}, s \models p$ iff $p \in V(s)$.
- $\mathcal{K}, s \models \phi \wedge \psi$ iff $\mathcal{K}, s \models \phi$ and $\mathcal{K}, s \models \psi$.
- $\mathcal{K}, s \models \neg\phi$ iff $\mathcal{K}, s \not\models \phi$.
- $\mathcal{K}, s \models K_i\phi$ iff for all s' such that $R_i(s, s')$, $\mathcal{K}, s' \models \phi$.
- $\mathcal{K}, s \models \neg K_i\phi$ iff there exists a state s' such that $R_i(s, s')$, $\mathcal{K}, s' \models \neg\phi$.
- $\mathcal{K}, s \models C\phi$ iff for all states s' such that $R^*(s, s')$ is true, $\mathcal{K}, s' \models \phi$
- $\mathcal{K} \models \phi$ iff $\mathcal{K}, s \models \phi$ for all states $s \in W$.

$R^*(s, s')$ is true if state s' is reachable from state s through any path of accessibility relations of any agent $j \in A$.

K	$K_i(\phi \to \psi) \to (K_i\phi \to K_i\psi)$
D	$K_i\phi \to \neg K_i \neg\phi$
T	$K_i\phi \to \phi$
4	$K_i\phi \to K_iK_i\phi$
5	$\neg K_i\phi \to K_i\neg K_i\phi$

Table 3.1: Epistemic axioms

The accessibility relations between states of a model $\mathcal{K}$, semantically, characterize the system in which $\mathcal{K}$ belongs. In this thesis we focus on the $\mathcal{S}5$system, which means that the accessibility relation is

- reflexive: if for all $s \in W$ and $i \in A$, we have $R_i(s, s)$
- symmetric: if $R_i(s, s')$ then $R_i(s', s)$
- transitive: if $R_i(s, s')$ and $R_i(s', s'')$, then $R_i(s, s'') \in R_i$

When an accessibility relation is reflexive, symmetric and transitive, it is called an *equivalence* relation.

In Table 4.1 we can see five axioms relating to epistemic logic (Van Ditmarsch et al., 2007). Axiom **D** states that agents do not hold inconsistent knowledge, axiom **T** that agents do not hold false knowledge, axiom **4**, also called *positive introspection*, states that agents know what they know, and axiom **5**, also called *negative introspection*, states that agents know what they do not know.

Semantically, an important property of the epistemic axioms is that they correspond to algebraic properties of a Kripke model, i.e. an epistemic axiom is valid in a Kriple model if the accessibility relation of that model satisfies certain conditions like reflexivity, symmetry etc. For example, axiom **T** is valid in all Kripke models where the accessibility relation is reflexive, while axiom **4** is valid in the Kripke models which are transitive.

The $\mathcal{S}5$system we are interested in is the system where the axioms **K**, **T**, **4** and **5** hold.

Revisiting the Muddy Children problem

In Figure 3.1 we can see the Kripke model $\mathcal{K}$ of the Muddy Children problem *before* the father's announcement. Each node represents a state, and the tuple within represents the truth values of the propositional literals. Example, the node with $(1, 0, 1)$ stands for the state where m_a is true, m_b is false and m_c is true. Each state represents a *possible* combination of muddy children, while the accessibility relations show which states are indistinguishable for each agent. For example, we have that agent a cannot distinguish between states $(0, 0, 1)$ and $(1, 0, 1)$: both states agree on what a can see (that m_b is false and m_c is true) but she does not know if she is muddy or not (if m_a is true or false). We do not show the reflexive relations ($R_i(s, s)$), which are present for all states and for all agents.

The father's announcement is *"at least one of you is muddy"*, which can be written as $(m_a \vee m_b \vee m_c)$. After the public announcement it is common knowledge the fact that all states s where $M, s \not\models (m_a \vee m_b \vee m_c)$ are not to be considered as possible, thus removing the state $(\neg m_a, \neg m_b, \neg m_c)$ which in Figures 3.1 is represented with $(0, 0, 0)$. This announcement results in Figure 3.2.

Given the Kripke model $\mathcal{K}$, we can evaluate formulas based on Definition 3.2.2:

- *Does Ann know she is muddy in the situation where only she is muddy?* is the same as checking whether $\mathcal{K}, s_1 \models K_a m_a$, where $s_1 = (1, 0, 0)$. The only accessibility relation in R_a involving s_1 is $R_a(s_1, s_1)$, and we have $\mathcal{K}, s_1 \models m_a$. In other words, in all the states Ann cannot distinguish from s_1, she is muddy - thus, she knows she is muddy.
- *In the situation where only Bob is muddy, does Ann know that Cadie knows that Bob is muddy?* is the same as checking whether $\mathcal{K}, s_2 \models K_a K_c m_b$, where $s_2 = (0, 1, 0)$. Ann cannot distinguish between s_2 and s_4, which means that it has to be $\mathcal{K}, s_2 \models K_c m_b$ and $\mathcal{K}, s_4 \models K_c m_b$ for $K_a K_c mb$ to be true in s_2.
  - Cadie cannot distinguish between s_2 and s_6, and we have that $\mathcal{K}, s_2 \models m_b$ and $\mathcal{K}_s, 6 \models m_b$, so $\mathcal{K}, s_2 \models K_c m_b$.
  - Cadie cannot distinguish between s_4 and s_7, and we have that $\mathcal{K}, s_4 \models m_b$ and $\mathcal{K}_s, 7 \models m_b$, so $\mathcal{K}, s_4 \models K_c m_b$.

  which means $\mathcal{K}, s_2 \models K_a K_c m_b$.

3.2.4. Dynamic Epistemic Logic

So far, we introduced epistemic logic as static: a formalization that allows us, given a situation at a fixed moment in time, to evaluate whether some epistemic formula ϕ is true or not at that moment in that situation. In real-world applications, agents interact

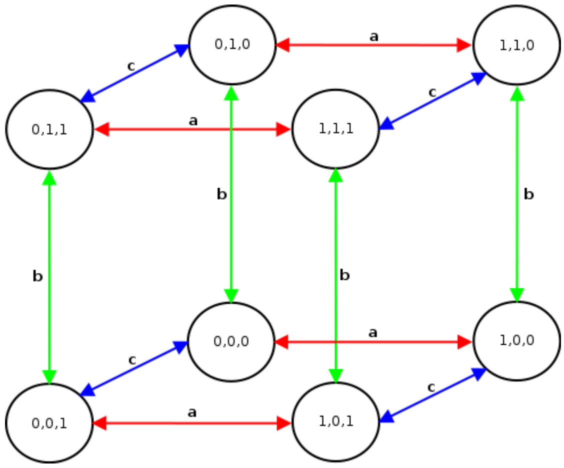

Figure 3.1: The Kripke model $\mathcal{K}$ of the Muddy Children problem *before* the father's announcement.

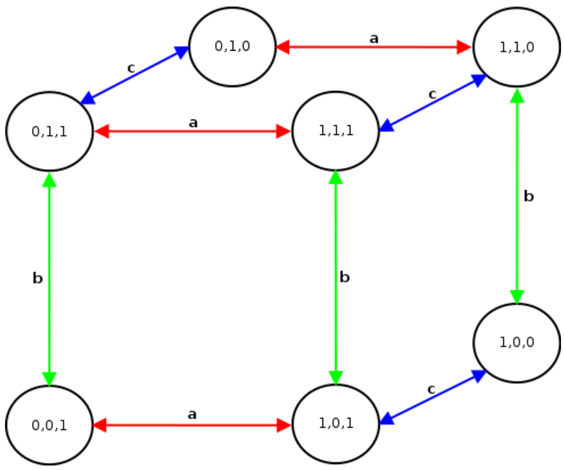

Figure 3.2: The Kripke model of the Muddy Children problem *after* the father's announcement.

with each other, either through their actions or by communicating information.

In the single-agent planning problems we have seen so far, there were two types actions: (i) the world-altering actions (or ontic actions or physical actions) which when applied change the environment (an agent moving, picking up an object, opening a door etc), and (ii) in the case of contingent planning problems, the sensors, which when triggered provide information to the agent.

In the presence of multiple agents, both acting and sensing extend their effect from just the environment and the beliefs an agent has, to the other agents' perception of the environment and their beliefs. As an example, if agent i publicly (as in *"it becomes common knowledge that"*) senses the truth value of p then:

- Agent i knows the truth value of p: $K_i p \vee K_i \neg p$, and
- All agents j know that he knows: $K_j(K_i p \vee K_i \neg p)$

Dynamic epistemic logic (DEL) is an umbrella term: it includes logics that deal with information change. Such logics extend the static formalization to one which incorporates how the agents' knowledge change due to physical, sensing and communicating actions (Gerbrandy and Groeneveld, 1997; Baltag et al., 1998; Baltag, 2000; Baltag and Moss, 2004a).

The intuition is that such events (sensing, communication, physical actions) can be modeled in a way similar to the one we use for modeling situations (Baltag et al., 1998; Baltag and Moss, 2004a; Van Ditmarsch et al., 2007). Given a Kripke model and an action model representing the applied event, we can produce a new Kripke model representing the new situation that occurs after the application of the action.

Definition 3.2.3. An action model is a structure $Q = \langle E, E^A, pre \rangle$, where

- E is a set of events,
- E^A is a set of accessibility relations $R_i^A \subseteq E \times E$, for every $i \in A$,
- $pre : E \to \mathcal{L}$, a precondition function s.t. $pre(e) \in \mathcal{L}$ for $e \in E$.

Constructing action models is similar to constructing Kripke models: identify how many events are possible and the perspective of each agent for each event (is he aware it occured? Can he distinguish it from another event? Does he know whether another agent is aware of it? etc).

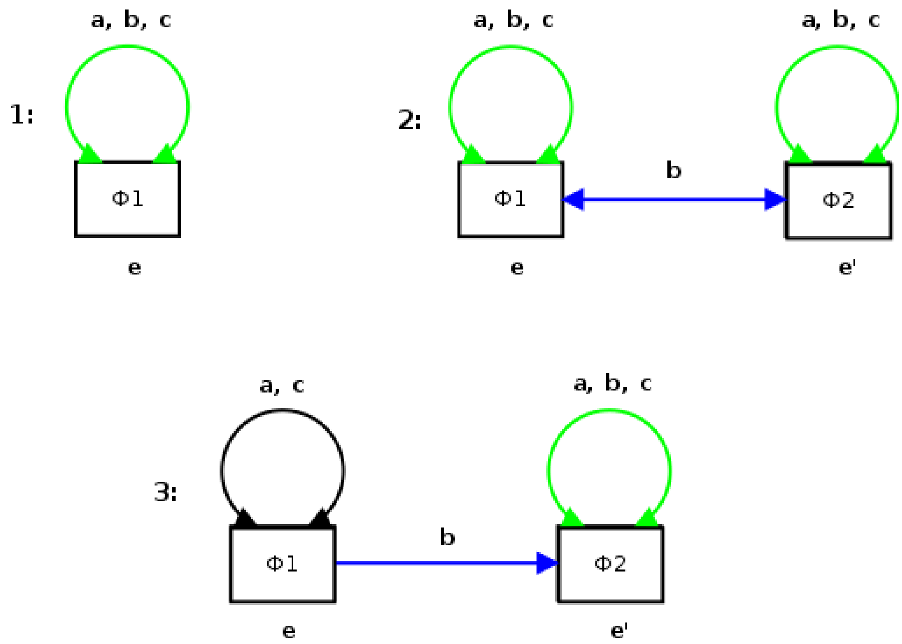

Figure 3.3: Examples of action events for the Muddy Children problem.

The action model we defined cannot model physical actions, but only actions that change the agents' information about the worlds and about other agents. Nevertheless, there are ways to extend it in order to include physical actions, as has been done in (Van Benthem et al., 2006; van Ditmarsch and Kooi, 2008a; van Ditmarsch et al., 2005; Baral et al., 2012). The main idea of the approaches is to add a set $post$ of effects (switch the truth value of a literal or define it) for each event $e \in E$, which will be applied only on states where the e is applicable.

Product update

Given a Kripke model and an action model, the *product update* is a method for obtaining a new Kripke model, representing the situation that occurs after applying the action model to the original Kripke model.

Definition 3.2.4. (Product Update) Given a Kripke model $\mathcal{K} = \langle W, R, V\rangle$ and an action model $Q = \langle E, E^A, pre\rangle$, we obtain a Kripke model $\mathcal{K}^Q = \langle W^Q, R^Q, V\rangle$ where:

- the worlds W^Q are the pairs $(s, e) \in W \times E$ such that $\mathcal{K}, s \models pre(e)$

- for each agent i, there exists $R_i^Q(s_e, s'_{e'})$ iff $R_i^A(s, s') \in R^A$ and $E_i^A(e, e') \in E^A$.
- for the truth values of propositions in the new states, we have $V^Q((s, e)) = V(s)$, for all worlds (s, e).

Similar to pointed Kripke models $(\mathcal{K}, s)$, we have pointed action models (Q, e). In this case, the result would be a pointed Kripke model $(\mathcal{K}^Q, (s, e))$.

In Figure 3.3 (1 to 3), we can see different actions events, defined over the Muddy Children problem. Events are represented with squares so that they are distinguished from the states of Kripke models. Figure 3.3.1., where $\Phi_1 = (m_a \vee m_b \vee m_c)$, is the action event of the father's announcement. All children hear the same announcement, and the action event is the one which takes us from the Kripke model in Figure 3.1 to the Kripke model in Figure 3.2. The action event in Figure 3.3.2, where $\Phi_2 = \neg\Phi_1$, represents the situation where agents a and c get to know the truth value of Φ_1, and b knows this fact. As an example, suppose $\Phi_1 = K_c m_c$. We can interpret the action event as the case where c tells a whether he knows the truth value of m_c. Agent b will not be able to distinguish between states where $K_c m_c$ is true and state where it is not, but he knows that a can. In the third example, suppose Φ_1 is some formula, and Φ_2 is just the statement true. In this case, the action event can be interpreted as the case where a and c get to know Φ_1, while b is not aware of that and believes that nothing has occurred.

We can now define a language for dynamic epistemic logic (Van Ditmarsch et al., 2007).

Definition 3.2.5. Given a set of atoms F and a set of agents A, the language $\mathcal{L}_d$ with common knowledge is defined by:

$$\phi ::= p \mid \neg\phi \mid (\phi \wedge \phi) \mid K_i\phi \mid C\phi \mid [\pi]\phi$$
$$\pi ::= (Q, e) \mid (\pi \cup \pi)$$

where $p \in F$, $i \in A$, and (Q, e) is a pointed action model, as defined in 3.2.4. Intuitively, $[\pi]\phi$ is a *dynamic operator* and stands for "ϕ holds after the occurrence of the event π".

3.2.5. Complexity

Given a pointed Kripke model $(\mathcal{K}, w)$ and a formula $\phi \in \mathcal{L}_d$, the model checking problem is defined as the problem of evaluating whether $\mathcal{K}, w \models \phi$. The satisfiability problem, given a formula $\phi \in \mathcal{L}_d$, is defined as the problem of determining if there exists a pointed Kripke model $(\mathcal{K}, w)$ such that $\mathcal{K}, w \models \phi$. For public announcement logic, which is a fragment of dynamic epistemic logic (the action model contains only one event and the reflexive relation to that event includes all agents), the model checking problem is in P (with common konwledge) (Kooi and van Benthem, 2004) and the satisfiability problem is PSPACE-complete without common knowledge (EXPTIME-complete with common knowledge) (Lutz, 2006). Without common knowledge, the model checking problem of $\mathcal{L}_d$ is PSPACE-complete and the satisfiability problem of $\mathcal{L}_d$ is NEXPTIME-complete (Aucher and Schwarzentruber, 2013). There are no known results concerning the complexity of DEL with common knowledge.

3.2.6. Knowledge vs Belief

Based on the axioms in Table 4.1, we defined $\mathcal{S}5$to be the system with axioms **K**, **T**,**4** and **5**, which denotes a Kripke model where the accessibility relation is an *equivalence* relation. Axiom **T** is also called the *truth* axiom: an agent can only know things which are *true*. For systems where **T** does not hold, we are talking about *belief* instead of *knowledge*: it is possible to believe something which might be wrong. The **KD45** system is usually considered a system which suffices to model *belief*: axiom **T** has been dropped and **D** has been added, denoting that agents may believe certain things to be true while they are false, but they cannot believe contradictory things. **KD45** is characterized by accessibility relations which are serial, transitive and euclidean.

3.2.7. Discussion

In this dissertation, we will talk about *beliefs* of agents. This should *not* be read as belief in terms of epistemic logic. We will use the word *beliefs* to denote the fact that agents hold some uncertainty about the world, and as a term it is borrowed from the notion of *belief state* that we mentioned in Chapter 2. Concerning individual agents, the uncertainty they hold about the world defines their *knowledge* (and not *belief*) about the world: what they know to be true, is actually true.

We saw that (dynamic) epistemic logic offers formal frameworks that allow to (i) *model* a (complex) situation that represents the knowledge and beliefs of multiple agents, and (ii) compute the next situation after an action is applied through a state-transition function in the form of product update. There are two restrictions, concerning these formulations, which are of importance to automated planning. The first is that defining action models is not clear: modeling an action can be as difficult (and complex) as modeling the world. The second restriction is that there are no computational techniques that allow to automatically generate plans. Given a Kripke model and an action model we can compute the resulting situation, but given an initial Kripke model, a goal Kripke model and a set of action models, there are no computational approaches that will generate a sequence of action models that achieve the goal. In this work we are trying to bridge this gap by considering a fragment of dynamic epistemic logic and providing a mapping to classical planning problems which will allow us to take advantage of the computation techniques in classical planning.

PART II

Belief Representations And Translations

CHAPTER **4**

Linear Multi-agent Planning

4.1. Introduction

Single-agent planning in partially observable settings is a well understood problem and existing planners can represent and solve a wide variety of meaningful instances. In the most common formulation, single-agent planning in partially observable environments is cast as a non-deterministic search problem in belief space where the beliefs are sets of states that the agent regards as possible (Bonet and Geffner, 2000). The work in partially observable or contingent planning has been focused on ways for representing beliefs and selecting actions (Bertoli et al., 2001; Brafman and Hoffmann, 2004; Albore et al., 2009; To et al., 2011; Brafman and Shani, 2012a).

Current approaches for representing beliefs in multi-agent dynamic settings, on the other hand, are based on Kripke structures (Fagin et al., 1995). multi-agent Kripke structures are triplets defined by a set of worlds, accessibility relations among the worlds for each of the agents, and truth valuations that define the propositions that are true in each world. While a truth valuation determines the *objective* formulas that are true in a world, the accessibility relation among worlds provides the truth conditions for *epistemic* formulas.

Dynamic epistemic logics extend epistemic logics with the ability to deal with change (van Ditmarsch et al., 2007a; van Ditmarsch and Kooi, 2008b; Van Benthem, 2011). The standard approach relies on *event models* and *product updates* by which both the agent beliefs and the events are represented by Kripke structures, and the resulting beliefs are captured by a suitable cross product of the two (Baltag et al., 1998; Baltag and Moss, 2004b). Syntactically, axiomatizations have been developed to capture the valid inferences in such a setting, and a number of approaches have been developed to facilitate modeling and inference (Baral et al., 2012; Herzig et al., 2005). A simple form of planning, however, where an event sequence is sought to achieve a given goal formula, has been shown to be undecidable in dynamic epistemic logic (Aucher and Bolander, 2013), while decidable subsets have been identified as well (Löwe et al., 2011).

In this chapter, we build on the methods developed for representing beliefs in single-agent planning to introduce a simple but expressive formulation for handling beliefs in multi-agent settings. The resulting formulation deals with multiple agents that can act on the world (physical or ontic actions), and can sense either the state of the world (truth of objective formulas) or the mental state of other agents (truth of epistemic formulas). The formulation captures and defines a fragment of dynamic epistemic logics that is simple and expressive, but which does not involve event models or product updates, and has the same complexity of belief tracking in the single agent setting and can benefit from the use of similar techniques. We show indeed that the problem of computing *linear multi-agent plans* (Bolander and Andersen, 2011) can be actually compiled into a *classical planning problem*, using the techniques that have been developed for compiling conformant and contingent problems in the single agent setting (Palacios and Geffner, 2009; Brafman and Shani, 2012b).

The proposed formulation exploits certain conventions and restrictions. First, while the agents can have private information as they have private sensors, they are all assumed to start with a *common initial belief* on the set of worlds that are possible. Second, the effects of physical actions on the world are assumed to be *deterministic.* And third, the *sequence of events* (physical actions, sensing events, and public announcements) that can change the state of the world or the knowledge state of the agents, is *public* to all the agents. In the formulation it is crucial to distinguish between the event of sensing the truth value of an objective or epistemic formula, and the agent coming to know that the formula is true or false. While the sensing event is public, as when all agents know the sensor capabilities of the other agents, the actual information provided by these sensors is private. For example, in the muddy children problem (Fagin et al., 1995), every child i is assumed to be capable of sensing the truth value of the atoms m_j encoding whether child j is muddy for $j \neq i$, and every child knows that. Yet this doesn't mean that children have access to the truth values revealed by the sensors that are not their own. The formulation does imply however that agents know what the other agents may potentially know, as agents start with the same knowledge and then learn about the world or about other agents using sensing events that are public.[1]

4.2. Language

We consider planning problems $P = \langle A, F, I, O, N, U, G\rangle$ where A is a set of agent names, F is a set of atoms, I is the initial situation, O is a set of *physical actions*, N is a set of *sensing actions*, U is set of *public (action) updates*, and G is the goal. A *plan* for P, as in classical planning, is a *sequence of actions* for achieving the goal G from the initial situation described by I. The main differences to classical planning result from the uncertainty in the initial situation, and the beliefs of the *multiple agents* involved. In addition the actions may come from any of the sets O, N, or U. If we let S stand for the set of all possible truth-valuations s over the atoms in F and call such valuations *states*, we assume that I is an objective formula over F which denotes a non-empty set of possible initial states b_I. A physical action a in O denotes a *deterministic* state-transition function f_a that maps any state s into a state $s' = f_a(s)$. A (parallel) sensing action in N is a set of expressions of the form **sense**$[A_k](\phi_k)$, where A_k is a non-empty set of agent names and ϕ_k is an objective or epistemic formula over the atoms F and the knowledge modalities K_i for $i \in A$. The action updates in U are denoted by expressions of the form **update**(ϕ) where ϕ is a formula. Finally, each action a has a precondition $Pre(a)$, which like the goal G are formulas as well. The grammar of these formulas can be expressed as:

$$\phi = p \,|\, \neg\phi \,|\, (\phi \wedge \phi) \,|\, (\phi \Rightarrow \phi) \,|\, K_i\phi$$

where p is an atom in F, and i an agent in A.

We regard *plans* as linear sequences of actions (Bolander and Andersen, 2011), and call P a *linear multi-agent planning problem*. While many problems require *non-linear plans*, as it is the case in contingent planning, *linear plans* suffice for a number of non-trivial contexts and provide the basis for more complex forms of plans. These linear plans involve sensing, however, but like conformant plans, no conditional branching.

[1] The assumptions in the model have points in common with the finitary S5 theories (Son et al., 2014) and with the notion of "only knowing" (Levesque, 1990; Halpern and Lakemeyer, 2001).

4.3. Dynamics of Knowledge Updates

In order to define the belief representation and dynamics, let us represent the event sequences or plans σ over a problem P by sequences of the form $e(0), \ldots, e(t)$, where $e(k)$ is the event from P that occurs at time k. When convenient, we will assume that the agent names are positive numbers i, $i = 1, \ldots, m$ for $m = |A|$, or that they can be enumerated in this way.

The beliefs of all the agents at time step t, called also the joint belief, will be denoted as $B(t)$, and it is represented by a vector of *conditional beliefs* $B(s,t)$, where s is one of the possible initial states, $s \in b_I$; namely,

$$B(t) = \{B(s,t) \mid s \in b_I\} \; . \tag{4.1}$$

The conditional beliefs $B(s,t)$ represent the beliefs of all the agents at time t, *under the assumption that the true but hidden initial state is s*. The reason for tagging beliefs with possible initial states is that for a *fixed (hidden) initial state s, the evolution of the beliefs $B(s,t)$ after an arbitrary event sequence is deterministic*. These conditional beliefs $B(s,t)$ are in turn represented by tuples:

$$B(s,t) = \langle v(s,t), r_1(s,t), r_2(s,t), ..., r_m(s,t)\rangle \tag{4.2}$$

where $v(s,t)$ is the state of the world that results from the initial state s after the event sequence $e(0), \ldots, e(t-1)$, and $r_i(s,t)$ is the set of possible *initial* states $s' \in b_I$ that agent i cannot distinguish at time t from the actual initial state s. Note that s may be the true initial state, and yet the agents may not know about it. Indeed, initially, they only know that if s is the true initial state, it must be part of the initial common belief b_I.

More precisely, the initial beliefs $B(s,t)$ at time $t = 0$ are given by:

$$v(s,t) = s \text{ and } r_i(s,t) = b_I \tag{4.3}$$

for all agents i, meaning that under the assumption that the hidden initial state is s and that no events have yet occurred, the actual state is s and the set of possible initial states is b_I.

The belief $B(t+1)$ at time $t+1$ is a function of the belief $B(t)$ and event $e(t)$ at time t:

$$B(t+1) = \mathbf{F}(B(t), e(t)) \tag{4.4}$$

We express this function by defining how the *type of event* $e(t)$ at time t affects the state $v(s, t+1)$ and the relations $r_i(s, t+1)$ that define the belief $B(t+1)$ at time $t+1$.

Physical Actions: If $e(t) = \mathbf{do}(a)$ for action a denoting a state-transition function f_a, then the current state $v(s, t)$ associated with the hidden initial state s changes according to f_a, but the sets of initial states $r_i(s, t)$ that agent i regards as possible remain unchanged

$$v(s, t+1) = f_a(v(s, t)) \tag{4.5}$$
$$r_i(s, t+1) = r_i(s, t) \tag{4.6}$$

where the index i ranges over all the agents in A.

All the other event types affect instead the sets $r_i(s, t+1)$ but not the state $v(s, t+1)$ that is regarded as current given the assumption that s is the true initial hidden state. That is, for the following event types $v(s, t+1) = v(s, t)$.

Sensing: If $e(t) = [\mathbf{sense}[A_1](\phi_1), \ldots, \mathbf{sense}[A_l](\phi_l)]$ is a sensing action denoting the set of sensing expressions $\mathbf{sense}[A_k](\phi_k)$ done in parallel at time t, the current state given s does not change, but the set of possible initial states compatible with the hidden initial state s for agent i given by $r_i(s, t+1)$ becomes:

$$\{s' \mid s' \in r_i(s, t) \text{ and } B(t), s' \models \phi_k \text{ iff } B(t), s \models \phi_k\} \tag{4.7}$$

where k ranges over all the indices in $[1, l]$ such that A_k includes agent i. If there are no such indices, $r_i(s, t+1) = r_i(s, t)$. The expression $B(t), s \models \phi$ denotes that ϕ is true in the belief at time t conditional on s being the true hidden state. The truth conditions for these expressions are spelled out below.

Updates: If $e(t) = \mathbf{update}(\phi)$, $r_i(s, t+1)$ is

$$\{s' \mid s' \in r_i(s, t) \text{ and } B(t), s' \models \phi\} \ . \tag{4.8}$$

The intuition for all these updates is the following. *Physical actions* change the current state of the world according to their state transition function. *Sensing actions* do not change the world but yield information. More specifically, when agent i senses the truth value of formula ϕ at time t, the set of initial states $r_i(s, t+1)$ that he thinks possible under the assumption that the true initial state is s, preserves the states s' in $r_i(s, t)$ that agree with s on the truth value predicted for ϕ at time t. Finally, a *public update* ϕ preserves the possible initial states s' in $r_i(s, t)$ that predict the formula ϕ to be true, and rules out the rest. The conditions under which a possible initial state s predicts that a formula ϕ will be true at time t, and the conditions under which a formula ϕ is true at time t, are made explicit below. Physical, sensing, and update actions are *applicable* at time t only when their preconditions are true at t.

4.4. From $B(t)$ to Kripke Structures

A Kripke structure is a tuple $\mathcal{K} = \langle W, R, V \rangle$, where W is the set of worlds, R is a set of binary accessibility relations R_i on W, one for each agent i, and V is a mapping from the worlds w in W into truth valuations $V(w)$. The conditions under which an arbitrary formula ϕ is true in a world w of a Kripke structure $\mathcal{K} = \langle W, R, V \rangle$, written $\mathcal{K}, w \models \phi$, are defined inductively:

- $\mathcal{K}, w \models p$ for an atom p, if p is true in $V(w)$,
- $\mathcal{K}, w \models \phi \vee \psi$ if $\mathcal{K}, w \models \phi$ or $\mathcal{K}, w \models \psi$,
- $\mathcal{K}, w \models (\phi \Rightarrow \psi)$ if $\mathcal{K}, w \models \phi$ implies $\mathcal{K}, w \models \psi$,
- $\mathcal{K}, w \models K_i\phi$ if $\mathcal{K}, w' \models \phi$ for all w' s.t. $R_i(w, w')$, and
- $\mathcal{K}, w \models \neg\phi$ if $\mathcal{K}, w \not\models \phi$

A formula ϕ is valid in the structure $\mathcal{K}$, written $\mathcal{K} \models \phi$, iff $\mathcal{K}, w \models \phi$ for all worlds w in $\mathcal{K}$. The conditions under which a possible initial state s predicts the truth of a formula ϕ at time t, written $B(t), s \models \phi$, follow from replacing the belief $B(t)$ by the Kripke structure $\mathcal{K}(t) = \langle W^t, R^t, V^t \rangle$ defined by $B(t)$ where

- $W^t = \{s \mid s \in Poss(t)\}$,
- $R_i^t = \{(s, s') \mid \text{ if } s' \in r_i(s, t)\ \}$,
- $V^t(s) = v(s, t)$

In these expressions, $Poss(t)$ stands for the initial states that remain possible at t; $Poss(t) = \cup_{s \in b_I} \cup_{i=1,\ldots,m} r_i(s, t)$. The worlds w in the structure $\mathcal{K}(t)$ are the possible initial states $s \in b_I$ that have not been ruled out by the updates. The worlds that are accessible from a world s to the agent i are the possible initial states s' that are in $r_i(s,t)$. Last, the valuation associated to a world s in this structure is the state $v(s,t)$ that deterministically follows from the possible initial state s and the event sequence up to $t-1$. $B(t), s \models \phi$ is thus true when $\mathcal{K}(t), s \models \phi$ is true, and $B(t) \models \phi$ iff $\mathcal{K}(t) \models \phi$. It is simple to show that the accessibility relations $R_i(t)$ are reflexive, symmetric, and transitive, meaning that the valid formulas satisfy the axioms of the epistemic logic S5.

4.5. Examples

4.5.1. Selective Communication

Let a, b, and c be three agents in a corridor of four rooms (p_1, p_2, p_3 and p_4 from left to right). The agents can move from a room to a contiguous room, and when agent i communicates (tells) some information, all the agents that are in the same room or in a contiguous room, will hear what was communicated. For example, if agent i expresses in room p_3 his knowledge about q, all agents in rooms p_2, p_3 and p_4 will come to know it. We consider the problem where agent a is initially in room p_1, b in p_2, c in p_3, and a has to find out the truth value of a proposition q and let c know without agent b learning it. The planning problem is encoded as the tuple $P = \langle A, F, I, O, N, U, G \rangle$ where $A = \{a, b, c\}$, $F = \{q\} \cup \{p(x,i)\}$, $x \in A$, $i \in [1,4]$, $I = \{p(a,1), p(b,2), p(c,3)\} \cup D$, where D contains the formulas expressing that each agent is in a single room, U is empty, and the goal is

$$G = (K_c q \vee K_c \neg q) \wedge (\neg K_b q \wedge \neg K_b \neg q) \ .$$

The set of physical actions is $O = \{right, left\}$ affecting the location of agent a in the obvious way (the actions have no effects when they'd move the agent away from the four rooms).

The sensing actions in N are two: the first about a learning the value of q when in p_2, the other, about a expressing his knowledge regarding q, which translates into agents b and c learning this when they are close enough to a. The first sensing action is thus **sense**(a,q) with the *precondition* $p(a,2)$, and the second is

$$\begin{aligned}\textbf{tell}(a,q) \ : \ [&\textbf{sense}(b, \phi_b \Rightarrow K_a q), \textbf{sense}(b, \phi_b \Rightarrow K_a \neg q),\\ &\textbf{sense}(c, \phi_c \Rightarrow K_a q), \textbf{sense}(c, \phi_c \Rightarrow K_a \neg q)] \ ,\end{aligned}$$

where **tell**(a,q) is the abbreviation of the action that we will use, and ϕ_b is the formula expressing that agent b is at distance less than 1 from agent a; namely $\phi_b = \vee_{i,j}[p(a,i) \wedge$

$p(b, j)]$ for i and j in $[1, 4]$ such that $|i - j| \leq 1$. The formula ϕ_c is similar but with c instead of b.

Initially, b_I contains the two states s_1 and s_2 satisfying I, the first where q is true, and the second where it is false. The initial belief at time $t = 0$ is $B(t) = \{B(s_1, t), B(s_2, t)\}$, where $B(s_i, t) = \langle v(s_i, t), r_a(s_i, t), r_b(s_i, t), r_c(s_i, t)\rangle$, $i = 1, 2$, and $r_x(s, t) = b_I$ for $x \in A$ and $s \in b_I$. The shortest plan is

$$\textbf{do}(right), \textbf{sense}(a, q), \textbf{do}(right), \textbf{do}(right), \textbf{tell}(a, q) .$$

The first sensing action can be done because its precondition $p(a, 2)$ holds in $B(1)$, and as an effect it removes agent a's uncertainty regarding q making $r_a(s_1, 2) = \{s_1\}$ and $r_a(s_2, 2) = \{s_2\}$. Agent a then knows whether q is true or false, and in principle, he could communicate this from his current location p_2 by performing the action **tell**(a, q) right away. But since the condition ϕ_b is true, b would come to know whether q is true, making the problem goal G unachievable. The effect of the two $right$ actions is to make $p(a, 4)$ true, and all other $p(a, i)$ atoms false, thus making the formula ϕ_b false and the formula ϕ_c true (i.e., agent a is now near c but not near b). The final event in the plan makes the truth value of q known to agent c but not to agent b, thus achieving the goal G. The first part follows because the state $v(s_1, 5)$ where agent a is at p_4 and q is true, makes the formula $\phi_c \Rightarrow K_a q$ sensed by agent c true, while the state $v(s_2, 5)$ makes this formula false, and similarly, the state $v(s_2, 5)$ makes the formula $\phi_c \Rightarrow K_a \neg q$ sensed by agent c true, while the state $v(s_1, 5)$ makes it false. As a result, the state s_2 is not in $r_c(s_1, 5)$, the state s_1 is not in $r_c(s_2, 5)$, both sets become singletons, and hence, the truth value of q becomes known to agent c. The same reasoning does not apply to agent b because the condition ϕ_b is false in the two states $v(s_1, 5)$ and $v(s_2, 5)$, and hence, both states trivially satisfy the formulas $\phi_b \Rightarrow K_a q$ and $\phi_b \Rightarrow K_a \neg q$ that are sensed by agent b, so that $r_b(s_1, 5)$ and $r_b(s_2, 5)$ remain unchanged, and equal to b_I.

4.5.2. Collaboration through Communication

As a third example, we consider a scenario where two agents volunteer information to each other in order to accomplish a task faster that would otherwise be possible without information exchange. It is inspired in the BW4T environment, a proposed testbed for joint activity (Johnson et al., 2009). There is a corridor of four rooms, p_1, p_2, p_3 and p_4 as in the previous example, four blocks $b_1, \ldots, b_4$ that are in some of the rooms, and two agents a and b that can move back and forth along this corridor. Initially, the two agents are in p_2 and do not know where the blocks are (they are not in p_2). When an agent gets into a room, he can see which blocks are in the room if any. The goal of the planning problem is for agent a to know the position of block b_1, and for agent b to know the position of block b_2. A shortest plan for the problem involves six steps: one agent, say a, has to move to p_1, the other agent has to move to p_3, they both must sense which blocks are in these rooms, and then they must exchange the relevant information. At that point, the goal would be achieved whether or not the information exchanged explicitly conveys the location of the target blocks. Indeed, if agent a does not see block b_1 in p_1 and agent b doesn't see this block either at p_3, agent a will then know that block b_1 must be in p_4 once b conveys to a the relevant piece of information; in this case $\neg K_b in(b_1, p_3)$.

The planning problem is $P = \langle A, F, I, O, N, U, G \rangle$, where $A = \{a, b\}$, $F = \{at(x, p_k), in(b_i, p_k)\}$, $x \in A$ $i, k \in [1, 4]$, $I = \{at(a, p_2), at(b, p_2)\} \cup D$, where D contains the formulas expressing that each block has a unique location. The set of updates U is empty, the goal is $G = (\vee_{k=1,4} K_a at(b_1, p_k)) \wedge (\vee_{k=1,4} K_b at(b_2, p_k))$, the actions in O are $right_x$ and $left_x$, for each agent $x \in A$ with the same semantics as in the example above, while the sensing actions are **sense**$(x, [in(b_1, p_k), \ldots, in(b_4, p_k)]$ with precondition $at(x, p_k)$ by which agent $x \in A$ finds out in parallel which blocks b_i, if any, are and are not in p_k, and **sense**$(x, [K_y in(b_i, p_k]))$, by which agent y communicates to agent $x \neq y$, whether he knows $in(b_i, p_k)$, $i, k \in [1, 4]$. There are thus four physical actions, eight actions that sense the world, and thirty-two communication actions. A shortest plan is:

do$(left_a)$, **do**$(right_b)$, **sense**$(a, [in(b_1, p_1), \ldots, in(b_4, p_1)])$,
sense$(b, [in(b_1, p_3), \ldots, in(b_4, p_3)])$,
sense$(a, K_b in(b_1, p_3))$, **sense**$(b, K_a in(b_2, p_1))$.

This sequential plan achieves the goal in spite of the uncertainty of the agents about the world and about the beliefs of the other agents.

4.6. Translation into Classical Planning

We show next how a *linear multi-agent planning problem* P can be compiled into a classical planning problem $K(P)$ such that the plans for P are the plans for $K(P)$. The language for $K(P)$ is STRIPS extended with *negation*, *conditional effects*, and *axioms*. This is a PDDL fragment supported by several classical planners. We will use $\neg L$ for a literal L to stand for the complement of L, so that $\neg\neg L$ is L. A conditional effect is an expression of the form $C \rightarrow E$ associated with an action a that states that the head E becomes true when a is applied and C is true. We write such effects as $a : C \rightarrow E$ when convenient. In addition planners normally assume that C and E are sets (conjunctions) of literals. If $C, C' \rightarrow E$ is one such effect, we take $C, \neg C' \rightarrow E$ as a shorthand for the effects $C, \neg L \rightarrow E$ for each literal L in C'. *Axioms* allow the definition of new, derived atoms in terms of primitive ones, called then the primitive fluents. The derived fluents can be used in action preconditions, goals, and in the body of conditional effects. While it's possible to compile axioms away, there are benefits for dealing with them directly in the computation of heuristics and in state progression (Thiébaux et al., 2005).

For mapping the multi-agent problem $P = \langle A, F, I, O, N, U, G\rangle$ into the classical problem $K(P)$, we will make some simplifying assumptions about the types of formulas that may appear in P. We will assume as in planning, and without loss of generality, that such formulas correspond to conjunctions of literals, where a literal L is an (objective) atom p from F or its negation, or an epistemic literal K_iL or $\neg K_iL$ where L is a literal and i is an agent in A. Other formulas, however, can easily be accommodated by adding extra axioms to $K(P)$. We will denote the set of objective literals in P by $L_F(P)$; i.e., $L_F(P) = \{p, \neg p | p \in F\}$, and the set of positive epistemic literals appearing in P by $L_K(P)$; i.e., $L_K(P)$ is the set of K_iL literals that appear as subformula of an action precondition, condition, goal, or sensing or update expression. Indeed, while the set of K_iL literals is infinite, as they can be arbitrarily nested, the set of such literals appearing in P is polynomial in the size of P. As an example, if $\neg K_2K_1\neg K_3p$ is a goal, then $L_K(P)$ will include the (positive epistemic) literals K_3p, $K_1\neg K_3p$ and $K_2K_1\neg K_3p$.

The translation $K(P)$ comprises the fluents L/s for the objective literals L in $L_F(P)$, and possible initial states $s \in b_I$, and fluents $D_i(s, s')$ for agents $i \in A$. The former express that the objective literal L is true given that s is the true initial state, while the latter that agent i can distinguish s from s' and vice versa. The epistemic literals K_iL appearing in P, such as K_3p, $K_1\neg K_3p$ and $K_2K_1\neg K_3p$ above, are mapped into derived atoms in $K(P)$ through the use of axioms. The expression C/s where C is a conjunction of literals L stands for the conjunction of the literals L/s.

Definition 4.6.1. Let $P = \langle A, F, I, O, N, U, G\rangle$ be a *linear multi-agent planning problem*. Then the translation $K(P)$ of P is the *classical planning problem with axioms* $K(P) = \langle F', I', O', G', X'\rangle$ where

- $F' = \{L/s : L \in L_F(P), s \in b_I\} \cup \{D_i(s, s') : i \in A, s, s' \in b_I\}$,
- $I' = \{L/s : L \in L_F(P), s \in b_I, s \models L\}$,
- $G' = G$,
- $O' = O \cup N \cup U$; i.e., same set of actions a with same preconditions $Pre(a)$, but with
  - effects $a : C/s \to E/s$ for each $s \in b_I$, in place of the effect $a : C \to E$ for physical actions **do**(a), $a \in O$,
  - effects $a : C/s, \neg C/s' \to D_i(s, s'), D_i(s', s)$ for each pair of states s, s' in b_I and (parallel) sensing actions a in N that involve a **sense**(i, C) expression, and

- effects $a : \neg C/s' \rightarrow D_i(s, s')$ for each pair of states s, s' in b_I and $i \in A$, for actions a of the form **update**(C),

- X' is a set of axioms:

  - one for each *positive derived fluent* $K_i L/s$ where $K_i L \in L_K(P)$ and $s \in b_I$ with (acyclic) definition $L/s \wedge \wedge_{s' \in b_I}[L/s' \vee D_i(s, s')]$,

  - one for each literal L in $L_F(P) \cup L_K(P)$ with definition $\wedge_{s \in b_I}[L/s \vee D_i(s, s)]$

In words, the primitive fluents in $K(P)$ represent the truth of the literals L in P conditioned on each possible hidden initial state s as L/s, and the (in)accessibility relation $D_i(s, s')$ among worlds. Initially, the worlds are all accessible from each other and $D_i(s, s')$ is false for all such pairs. On the other hand, L/s is true initially if L is true in s. The goal G' of $K(P)$ is the same as the (conjunctive) goal G of P, and the actions O' in $K(P)$ are the actions in the sets O, N, and U of P with the same preconditions. However, in the translation, the effect of physical actions is on the L/s literals, while the effect of sensing actions and updates is on the $D_i(s, s')$ literals, with the literals $D_i(s, s)$ for any i being used to denote that the world s is no longer possible. Last, the truth conditions for epistemic literals in the translation is expressed by means of axioms in terms of the primitive literals L/s and $D_i(s, s')$.

The complexity of the translation is quadratic in the number $|b_I|$ of possible initial states. Its soundness and completeness properties can be expressed as follows:

Theorem 1. An action sequence π is a plan that solves the linear multi-agent planning problem P iff π is a plan that solves the classical planning problem with axioms $K(P)$.

The translation above follows the pattern of other translations developed for conformant and contingent planning problems in the single agent setting (Palacios and Geffner, 2009; Albore et al., 2009; Brafman and Shani, 2012a) and is closest to the one formulated by Brafman and Shani (2012b). Actually, Brafman, Shani and Zilberstein have recently developed a translation of a class of multi-agent contingent planning problems that they refer to as Qualitative Dec-POMDPs (Brafman et al., 2013), as it's a "qualitative" (logical) version of Decentralized POMDPs (Bernstein et al., 2000). A key difference with our linear multi-agent planning problems is that in Q-Dec-POMDPs the agents have beliefs about the world, but not about each other. Hence there are no epistemic modalities or epistemic formulas.

Problems	#Atoms	#Actions	#Axioms	#States	A*(max)	A*(cea)	BFS(add)	FF-X
MuddyChildren(3)	212	5	72	8	(0.02 - 0.01) / 6	(0.02 - 0.02) / 6	(0.02 - 0.02) / 6	0.01 / 6
MuddyChildren(4)	816	6	192	16	(0.16 - 0.06) / 8	(0.1 - 0.01) / 8	(0.15 - 0.02) / 8	0.1 / 8
MuddyChildren(5)	3312	7	480	32	(1.64 - 1.1) / 10	(0.7 - 0.1) / 10	(0.8 - 0.22) / 10	3.6 / 10
MuddyChildren(6)	14080	8	1152	64	(24.5 - 20.1) / 12	(5.4 - 1.1) / 12	(8 - 3.3) / 12	87 / 12
MuddyChildren(7)	61504	9	2688	128	(360 - 311) / 14	(55.1 - 9) / 14	(109.8 - 64) / 14	–
Collab-and-Comm(2)	348	22	132	9	(0.1 - 0.04) / 6	(0.06 - 0.02) / 6	(0.06 - 0.02) / 6	0.05 / 8
Collab-and-Comm(3)	1761	28	546	27	(1.6 - 1.1) / 6	(0.8 - 0.25) / 6	(0.85 - 0.25) / 6	9.3 / 8
Collab-and-Comm(4)	10374	34	2112	81	(48.1 - 33) / 6	(20.3 - 5.3) / 6	(22 - 6.5) / 6	765 / 8
Selective-Comm	59	7	20	2	(0.01 - 0.01) / 9	(0.01 - 0.01) / 9	(0.01 - 0.01) / 9	0.01 / 9
MuddyChild(3,1)	180	6	40	8	(0.01 - 0.01) / 5	(0.01 - 0.01) / 5	(0.01 - 0.01) / 5	0.01 / 5
MuddyChild(4,1)	720	8	96	16	(0.1 - 0.02) / 7	(0.1 - 0.01) / 7	(0.1 - 0.02) / 7	0.05 /7
MuddyChild(5,2)	3056	10	224	32	(1.3 - 0.06) / 8	(1.14 - 0.02) / 8	(1.2 - 0.06) / 8	1.75 / 8
MuddyChild(5,1)	3056	10	224	32	(1.3 - 0.08) /9	(1.14 - 0.02) / 9	(1.2 - 0.08) / 9	1.82 / 9
MuddyChild(6,2)	13440	12	512	64	(23 - 0.6) / 10	(22.1 - 0.2) / 10	(22.6 - 0.7) / 10	50 / 10
MuddyChild(6,1)	13440	12	512	64	(23 - 0.6) / 11	(22.1 - 0.25) / 11	(22.7 - 0.7) / 11	51.5 / 11
MuddyChild(7,2)	59968	14	1152	128	(554.5 - 4.5) / 12	(551 - 1.5) / 12	(555 - 5.7) /12	–
Sum(3)	306	10	90	9	(0.02 - 0.01) / 3	(0.02 - 0.01) / 3	(0.04 - 0.02) / 3	0.02 / 3
Sum(4)	963	13	234	18	(0.32 - 0.2) / 5	(0.2 - 0.02) / 5	(0.2 - 0.06) / 5	0.6 / 5
Sum(5)	2325	16	480	30	(26.5 - 26) / 7	(0.7 - 0.1) / 7	(0.8 - 0.25) / 7	9.1 / 7
Sum(6)	4770	19	855	45	–	(2.4 - 0.7) / 10	(3.2 - 1.5) / 10	53 / 10
Sum(7)	8757	22	1386	63	–	(7.5 - 2.9) / 11	(9.5 - 5.3) / 11	241 / 13
WordRooms(25,8)	935	56	535	8	(9.4 - 9.3) / 9	(0.25 - 0.1) / 11	(0.25 - 0.1) / 11	6.2 / 11
WordRooms(25,10)	1183	56	663	10	(18 - 17.8) / 9	(0.5 - 0.2) / 11	(0.5 - 0.2) / 11	11.9 / 11
WordRooms(25,12)	1439	56	791	12	(60 - 59.6) / 10	(0.6 - 0.26) / 14	(0.6 - 0.3) / 14	20.3 / 10
WordRooms(30,14)	1913	56	1059	14	(134.3 - 133.7) / 10	(1.1 - 0.5) / 15	(1.1 - 0.5) / 15	49.2 / 14
WordRooms(30,16)	2215	56	1207	16	(207 - 206) / 10	(1.5 - 0.7) / 15	(1.5 - 0.6) / 15	73 / 16

Table 4.1: Experimental results. Problems P shown on the left. The columns indicate number of atoms, actions, and axioms in $K(P)$, the number of possible initial states for P, and the resulting times and plan lengths. FF-X refers to the version of FF that supports axioms. The other columns refer to three different configurations of Fast Downward using the same search algorithm A* and the heuristics h_{max}, h_{cea} and h_{add}. The first configuration yields provably shortest plans. In the FF-X column, X/Y stands for X seconds and plan length Y. For Fast Downward, X-Y/Z stands for X seconds of total time, Y seconds spent on the search, and plan length Z. Unsolved problems indicated as "–".

4.7. Experiments

We have tested the translation above by taking a number of problems P and feeding the translations $K(P)$ into classical planners. The results are shown in Table 4.1.[2] As classical planners we used the version of FF known as FF-X (Thiébaux et al., 2005) that supports axioms and is available from J. Hoffmann, and three configurations of Fast Downward (Helmert, 2006) in a version that we obtained from M. Helmert that does less preprocessing. The three configurations differ just on the heuristic that is used to guide an A* search which is *optimal* for the admissible h_{max} heuristic. The results have been obtained on a Linux machine running at 2.93 GHz with 4 GB of RAM and a cutoff of 30 minutes.

[2]Software and data at http://www.dtic.upf.edu/∼fkominis/

A couple of optimizations have been implemented in the translation $K(P)$. In particular, we take advantage of the symmetry of the $D_i(s, s')$ predicates to reduce these atoms in half. In addition, for sensing actions **sense**(i, C) where C is a *static* objective atom, we define the effects *unconditionally* for all pairs $s, s' \in b_I$ such that s and s' disagree on the truth value of C.

About the list of domains in the table, the first three have been discussed already: MuddyChildren(n) with n children, Collab-through-Comm(n) with n blocks, (only two blocks are relevant though), and Selective-Communication. The new domains are discussed below.

4.7.1. Active Muddy Child

MuddyChild(n, m) is a reformulation of MuddyChildren where a particular child must find out whether he is muddy or not. For this he can ask individually each other child i whether i knows that he is muddy, with all other children listening the response. Thus, while in MuddyChildren(n) there is just one epistemic sensing action that lets every child know whether each child knows that he is muddy, in MuddyChild(n, m), there are $n - 1$ epistemic actions depending on the child being asked. In addition, to make things more interesting, the goal in MuddyChild(n, m) is for the selected child k to find out whether he is muddy, *given* that m of the children are *not* muddy in the actual world. For example, in MuddyChild($5, 2$), this goal can be encoded by the formula $(\neg m_1 \land \neg m_2) \supset (K_3 m_3 \lor K_3 \neg m_3)$. The result of this conditional goal is that in the resulting (shortest) plans, child 3 will not ask questions to children 1 and 2, as there is nothing to achieve in the worlds where either one of them is muddy. While this is not initially known, the child has physical sensors to discover that. Actually, in this domain, in order to represent the initial situation where the children have received the father's announcement and the information from their physical sensors, we force on all plans an initial sequence of actions that contain these $n + 1$ actions. This is easy to do by adding extra fluents. The shortest plans for MuddyChild(n, m) thus will involve these $n + 1$ actions followed by $n - m - 1$ epistemic actions.

4.7.2. Sum

Sum(n) is a domain based on "What is the Sum?" (van Ditmarsch et al., 2007b), which in turn borrows from the "Sum and Product Riddle" (van Ditmarsch et al., 2008) and the Muddy Children. There are three agents a, b, and c, each one with a number on his forehead between 1 and n. It is known that one of the numbers must be the sum of the other two. In addition, each agent can see the numbers on the other agent's foreheads, and can be asked to publicly announce whether he knows that he has a specific number. The goal is for one selected agent or two to learn their numbers. Atoms x_i, for $x \in A = \{a, b, c\}$ and $1 \leq i \leq n$ are used for indicating that agent x has the number i on his forehead. We use one action that lets all agents know the numbers on the forehead of the other agents in parallel. In addition, there are $3n$ actions that let all agents sense whether agent x knows that he has the number i, $x \in A$ and $1 \leq i \leq n$.

The problem is subtle. Consider for example the smallest problem with $n = 3$ where agent a must learn his number, i.e., $G = K_a a_1 \vee K_a a_2 \vee K_a a_3$. Since the largest number must be the sum of the other two, and hence must be larger than the other two, these two other numbers can be 1 and 1, or 1 and 2. There are thus two different tuples of numbers that are possible, $1, 1, 2$ and $1, 2, 3$, to be distributed among the 3 agents, resulting into 9 possible (initial) states and $|b_I| = 9$.

If agent a sees that a second agent has the number 3, he will know his number from looking at the third agent: if he has number 2, then a must have number 1, and if the third agent has number 1, a must have number 2. On the other hand, if a sees only numbers 1 and 2, he will not know whether he has number 1 or 3. Yet he can ask the agent with the number 1 whether he knows that he has the number 1: if he knows, then a knows that he has number 3, else, he has number 1. These various scenarios can be obtained by setting the goal to an implication like $\neg a_3 \supset K_a a_1 \vee K_a a_2$. The goals for the instances in the table do not involve conditions on the actual world and thus must work *for all the worlds that are possible*.

In Sum(3), the goal is for one agent, say a, to learn his number and the plan involves all agents sensing the numbers of the others in parallel, and then b and c reporting in sequence whether they each know that his own number is 1. The total number of actions in the plan is thus 3. There are three cases to consider to show that the plan works. If the report from b is $K_b b_1$, a and c must have the numbers 2 and 3, or 3 and 2, but since a can see c, he can figure out his number. Let us thus assume that the report from b is $\neg K_b b_1$ followed by c reporting $K_c c_1$. In such a case, from the first observation, agents a and c cannot have 2 and 3, or 3 and 2, and from the second, a_1 and b_1 cannot be both true either. Thus a and b must have the numbers 2 and 1, 2 and 3, or 3 and 2. Once again, since a can see b, a can figure out his number. Last, if the sensing results in $\neg K_b b_1$ followed by $\neg K_c c_1$, a and b must have the numbers 1 and 1, 1 and 2, or 1 and 3. Therefore a will be able to know that his number is 1.

Interestingly, there is *no* plan for the goal when all agents must learn their numbers. Let us assume that b reports first, and let us focus on two of the possible initial states where the numbers for a, b and c are 2,1,1 and 2,3,1 respectively. In state 2,1,1, a will know his number, and b will express ignorance, from which c will learn that his number is 1. Agent b can predict this, and hence will not learn anything else from either a or c. Thus, the first agent that speaks up in the *linear* plan, won't be able to figure out his number in all states.

4.7.3. Word Room

WordRoom(m, n) is a variation of the collaboration through communication example. It involves two agents a and b that must find out a hidden word from a list of n possible words. The words can have at most 7 letters with the i-th letter of the word being in room r_i, $i = 1, \ldots, 7$. The two agents can move from a corridor to each of the rooms, and from any room back to the corridor. There is no direct connection among rooms, the two agents cannot be in the same room, and they both start in the corridor. The agents have sensors to find out the letter in a room provided that they are in the room, and they can communicate the truth of the literals $K_x l_i$ where x is one of the two agents and l_i expresses that l is the i-th letter of the hidden word. The former amounts to 14 sensing actions of the form **sense**$(x, [l_i, l'_i, l''_i, \ldots])$ with the precondition that agent x is in room i, and where $l, l', \ldots$ are the different letters that may appear at position i of some of the n words. The parameter m in problem WordRoom(m, n) stands for the total number of l_i atoms. There are also 7 actions **sense**$(a, [K_b l_i, K_b l'_i, K_b l''_i, \ldots])$ where b communicates what he knows about room i to a, and similarly, 7 actions where a communicates to b. If we add the 14 actions for each agent moving from a room to the corridor and back, the total pool of actions is 56. The shortest plan for these problems is interesting when there is a lot of overlap among the n possible words, and in particular, when it may be more efficient for an agent to communicate *not* the letters that he has observed, but the letters that he can derive from what he knows.

4.8. Relation to Single Agent Beliefs and DEL

The proposed formulation for handling beliefs in a multi-agent setting sits halfway between the standard formulation of beliefs in single agent settings as found in conformant and contingent planning (Geffner and Bonet, 2013), and the standard formulation of beliefs in the multi-agent settings as found in dynamic epistemic logics (van Ditmarsch et al., 2007a; van Ditmarsch and Kooi, 2008b). In the single agent settings, beliefs are represented as the sets of states b that are possible, and physical actions a, whether deterministic or not, affect such beliefs deterministically, mapping a belief b into a belief $b_a = \{s \mid s \in F(a, s') \text{ and } s' \in b\}$ where F represents the system dynamics so that $F(a, s)$ stands for the set of states that may follow action a in state s. If the action a is *deterministic*, $F(a, s)$ contains a single state. The belief resulting from doing action a in the belief b and getting an observation token o is $b_a^o = \{s \mid s \in b_a \text{ such that } o \in O(a, s)\}$

where O represents the *sensor model* so that $O(a, s)$ stands for the set of tokens that can be observed after doing action a, resulting in the (possibly hidden) state s. Sensing is noiseless or *deterministic*, when $O(a, s)$ contains a single token. Interestingly, when both the actions and the sensing are *deterministic*, the set of beliefs $B'(t)$ that may follow from an initial belief b_I and a given action sequence is $B'(t) = \{b(s,t) \mid s \in b_I\}$ where $b(s,t)$ is the unique belief state that results from the action sequence and the initial belief state b_I when s is the hidden state. This expression has indeed close similarities with the beliefs $B(t)$ defined by (4.1) and (4.2) above.

While the proposed formulation is an extension of the belief representation used in single-agent planning, it represents also a *fragment* of *dynamic epistemic logics* where the Kripke structure $\mathcal{K}(t+1)$ that represents the belief at time $t+1$ is obtained from the Kripke structure $\mathcal{K}(t)$ representing the beliefs at time t and the Kripke structure representing the event at time t called the *event model*. The update operation is known as the *product update* as the set of worlds of the new structure is obtained by taking the cross product of the sets of worlds of the two time t structures. In particular, using the framework laid out in (van Ditmarsch and Kooi, 2008b; Bolander and Andersen, 2011) for integrating epistemic and physical actions, the basic actions in our language can be all mapped into simple event models. The event model for **do**(a) is given by a single event whose postcondition in a state s is $f_a(s)$. The event model for **update**(ϕ) has also a single event with precondition ϕ and null postcondition. Finally, the event model for **sense**(A, ϕ) has two events that can be distinguished by the agents in A but not by the other agents, one with precondition ϕ, the other with precondition $\neg\phi$, and both with null postconditions. While the proposed formulation captures only a fragment of dynamic epistemic logics, for this fragment, it provides *a convenient modeling language*, *a simple semantics*, and *a computational model.*

4.9. Conclusion

We have introduced a framework for handling beliefs in the multi-agent setting that builds on the methods developed for representing beliefs in single-agent planning. The framework also captures and defines a fragment of dynamic epistemic logics that does not require event models or product updates, and has the same complexity as belief tracking in the single agent setting (exponential in the number of atoms). We have also built on these connections to show how the problem of computing linear multi-agent plans can be mapped into a classical planning problem, and have presented a number of examples and experimental results.

A basic assumption is that all uncertainty originates in the set of states that are possible initially and hence that actions are deterministic. Still, non-deterministic physical and sensing actions can be introduced by reducing them to deterministic actions whose effects are made conditional on extra hidden variables. Similarly, while all agents are assumed to start with the same belief state, different initial beliefs that result from a common belief and different public sensing events can be handled easily as well.

CHAPTER 5

Online Planning and Dialogues

5.1. Introduction

Single-agent planning with partial observability is a hard computational problem where even the size of the required policies is often exponential in the problem size (Rintanen, 2004). For avoiding this bottleneck, *on-line* approaches have been developed that rather than computing full policies off-line, compute the next action to do given the observations gathered (Albore et al., 2009; Brafman and Shani, 2012b; Bonet and Geffner, 2014b).

In this chapter, we address the problem of on-line planning in partially observable environments in the presence of multiple agents that share a common goal and plan with beliefs that can be about the world or about the possibly nested beliefs of other agents. While this is a setting addressed by *dynamic epistemic logics* (van Ditmarsch et al., 2007a; van Ditmarsch and Kooi, 2008b; Van Benthem, 2011), we build on the formulation we presented in the previous chapter that captures a fragment of DEL, for which it provides a convenient modeling language, a simple semantics, and procedures akin to those used in the single-agent setting (Kominis and Geffner, 2015). In this approach, the basic assumptions are that physical actions are deterministic, all agents know the sensors available to each of the agents, and the set of possible initial states and actions that have been applied are common knowledge. There is a clear tradeoff between expressivity, simplicity, and computational efficiency, and other approaches addressing planning in a multi-agent setting make different tradeoffs (Baral et al., 2012; Brafman et al., 2013; Muise et al., 2015; Engesser et al., 2015; Cooper et al., 2016).

For using the previous chapter's formulation in the on-line setting, three issues need to be addressed. First, beliefs must take into account the actual observations gathered by the agents. Second, plans must be computed by the agents themselves using their own private information. And third, plans do not have to achieve the goal for all possible initial states, but for the true hidden initial state. We address these issues by adopting a suitable formulation of truth in the on-line setting that is used within a plan-execute-observe-and-replan cycle along with a translation into classical planning for selecting actions. The resulting on-line planning algorithm is guaranteed to reach the goal in a bounded number of calls to a classical planner provided that there are no dead-ends, even if different agents are chosen to plan in the different replanning episodes. We also show that interesting agent dialogues arise in this setting where agents request, provide, and volunteer information in a collaborative, goal-directed manner.

5.2. Motivation

We will use the Active Muddy Child problem of Section 4.7.1 for illustrating the differences between the off-line and on-line settings, where planning with epistemic goals is more crucial than in the standard partially observable setting of single-agent planning. While in the original puzzle, the father announces that at least one child is muddy, and then asks the children repeatedly whether they know whether they are muddy or not until the muddy children all infer that they are muddy, in the Active version, one of the children is the one asking the questions to find out whether he is muddy or not. Moreover, he has to ask these questions to one child at a time, whose answer, however, is heard by all the children. A *conformant plan* for the Active Muddy Child problem is one where the active child asks the question to each one of the children in turn without leaving any one out, in any order. The plan achieves the goal regardless of the true initial state. The problem has indeed $2^n - 1$ possible initial states where different subsets of children are muddy, excluding the state where no child is muddy that is common knowledge.

In the *on-line version* of the Active Muddy Child problem, the child asking the questions to figure out whether he is muddy or not, does not have to ask each of the children in turn whether they are muddy or not. The "planning child", like the other children, senses the world and can perfectly see which children are muddy and which ones are not, except for himself. A more effective strategy in the on-line setting is to approach *only* the children that are seen to be muddy. Any plan where the "planning child" asks the question to each of the children that he sees muddy, will achieve the goal.

The difference between the off-line and on-line setting is not the presence of observations that the planning agent can use for selecting actions. This is actually a result of a partially observable environment. The key difference is that in former the plans are supposed to work for *all possible initial states*, while in the latter they are supposed to work for *one possible initial state only:* the true hidden state. For example, the standard solutions to contingent planning problems are contingent trees. This form of contingent planning makes use of observations but it's an *off-line* method: the trees cover all the possibilities and hence all possible initial states. The solution form in *on-line* planning is not a tree but an *action sequence*, as the solution is supposed to work for one particular state: the true hidden state. In such a setting, the actual observations provide information about the hidden state, and hence, about the next actions to be done so that the planning agent will be certain that the goal is true. The planning agent in the on-line setting may find useful to consider many and even all possibilities before deciding what to do next. Yet, this is a criterion for choosing actions; the solution of the on-line problem is given by the sequence of actions taken if the actions and the observation let the agent know that the goal was reached.

The distinction between off-line and on-line planning is often left implicit and without formalization in the single-agent, partial observable setting, because goals in the latter are objective and refer to the world. In the on-line setting of epistemic, multi-agent planning, on the other hand, things are different and force us to make explicit and formal the conditions under which an epistemic goal is achieved from the internal perspective of the planning agent, and hence the role that the hidden true state plays in such conditions.

5.3. Language

We consider planning problems $P = \langle A, F, I, O, N, S, G \rangle$ where A is the set of agent names or indexes, F is the set of relevant atoms or fluents, I represents the initial situation in the form of an objective formula over F, O is the set of *physical actions*, N is the set of *sensing actions*, S is the set of (passive) *sensors*, and G is the goal. States represent truth-valuations over F, and the set of possible initial states b_I is made of the states that satisfy I. The physical actions a define a mapping f_a such that $f_a(s)$ represents the state that result from applying action a in the state s. Syntactically, such mappings are defined through a set of conditional effects of the form $C \rightarrow L$, where L is a literal and C is a formula over F or the atom $true$ that is normally omitted. A sensing action in N is a set of expressions of the form **sense**$[i](\phi)$, where i is an agent, and ϕ is an objective or epistemic formula. A result of the action is that the truth value of ϕ is revealed to agent i. A (parallel) sensing action in N is a set of expressions of the form **sense**$[A_k](\phi)$, where the truth of ϕ is revealed to all the agents $j \in A_k$. Unlike sensing actions, sensors reveal information without having to act. We denote passive sensors like sensing actions but with the letter "p" in front; namely, as **psense**$[i](\phi)$ and

psense$[A_k](\phi)$. Also, we write **sense**(ϕ) and **psense**(ϕ) when the sensing involves all the agents, i.e. $A_k = A$.

The goal G and the formulas ϕ above can be epistemic. The epistemic formulas ϕ include the atoms in F, and recursively, the formulas $K_i\phi$ for $i \in A$, and the boolean combinations of such formulas where K_i is the standard operator in logics of knowledge (Fagin et al., 1995).

Finally, physical actions a have a precondition formula $Pre(a)$ that can be objective or epistemic. We assume that each action has an 'owner" and that the action is applicable if the owner knows that the precondition is true (Engesser et al., 2015).

5.4. Beliefs

Beliefs are represented by a suitable collection of sets of states. The beliefs define a Kripke structure where arbitrary epistemic formulas can be evaluated.

5.4.1. External View

The beliefs of all the agents at time step t, denoted as $B(t)$, is represented by the beliefs $B(s,t)$ conditional on $s \in b_I$ being the true initial state, given as in Section 4.3:

$$B(s,t) = \langle v(s,t), r_1(s,t), r_2(s,t), ..., r_m(s,t)\rangle$$

where $v(s,t)$ is the state of the world that results from the initial state s after the action sequence $\pi(0), \ldots, \pi(t-1)$, and $r_i(s,t)$ is the set of possible *initial* states $s' \in b_I$ that agent i cannot distinguish at time t from the actual initial state s.

For $t = 0$, $v(s,t) = s$ and $r_i(s,t) = b_I$ for all agents i, while for $t > 0$, $B(t+1)$ is determined by $B(t)$ and the action $\pi(t)$ at time t.

If $\pi(t)$ is a sensing action or contains such actions, the current state given s does not change, i.e., $v(s,t+1) = v(s,t)$, but the set of possible initial states compatible with the hidden initial state s for agent i given by $r_i(s,t+1)$ becomes:

$$\{s' | s' \in r_i(s,t),\ B(t), s' \models \psi \text{ iff } B(t), s \models \psi, \forall\, \psi \in O_i(t)\}$$

where $O_i(t)$ represents the *observables* at time t and contains all the formulas ϕ such that the action **sense**$[A_k](\phi)$ is in $\pi(t)$ or **psense**$[A_k](\phi)$ is a passive sensor, in both cases with $i \in A_k$. The expression $B(t), s \models \phi$ denotes that ϕ is true in the belief $B(t)$

conditional on s being the true hidden state. The truth conditions for such expressions are spelled out below.

If $\pi(t)$ is a physical action a, the current state $v(s,t)$ associated with the hidden initial state s changes according to transition function f_a as $v(s,t+1) = f_a(v(s,t))$, while the sets of initial states $r_i(s,t)$ change according to the displayed formula above, where the observables in $O_i(t)$ result from the passive sensors only. In addition, if the action a is "owned" by agent j, states $s \in r_i(s,t+1)$ where $B(t), s \models K_j Pre(a)$ does not hold are removed from $r_i(s,t+1)$, meaning that agents i learn that action a is then applicable.

5.4.2. From Beliefs to Kripke Structures

A Kripke structure is a tuple $\mathcal{K} = \langle W, R, V \rangle$, where W is the set of worlds, R is a set of binary accessibility relations R_i on W, one for each agent i, and V is a mapping from the worlds w in W into truth valuations $V(w)$. The conditions under which an arbitrary formula ϕ is true in a world w of a Kripke structure $\mathcal{K} = \langle W, R, V \rangle$, written $\mathcal{K}, w \models \phi$, are defined inductively (Fagin et al., 1995):

- $\mathcal{K}, w \models p$ for an atom p, if p is true in $V(w)$,
- $\mathcal{K}, w \models \phi \vee \psi$ if $\mathcal{K}, w \models \phi$ or $\mathcal{K}, w \models \psi$,
- $\mathcal{K}, w \models (\phi \Rightarrow \psi)$ if $\mathcal{K}, w \models \phi$ implies $\mathcal{K}, w \models \psi$,
- $\mathcal{K}, w \models K_i\phi$ if $\mathcal{K}, w' \models \phi$ for all w' s.t. $R_i(w, w')$,
- $\mathcal{K}, w \models \neg\phi$ if $\mathcal{K}, w \not\models \phi$

A formula ϕ is valid in the structure $\mathcal{K}$, written $\mathcal{K} \models \phi$, iff $\mathcal{K}, w \models \phi$ for all worlds w in $\mathcal{K}$. The conditions under which a possible initial state s predicts the truth of a formula ϕ at time t, written $B(t), s \models \phi$, follow from replacing the belief $B(t)$ by the Kripke structure $\mathcal{K}(t) = \langle W^t, R^t, V^t \rangle$ defined by $B(t)$ where $W^t = \{s \mid s \in b_I\}$, $R_i^t = \{(s, s') \mid s' \in r_i(s,t)\ \}$, and $V^t(s) = v(s,t)$.

The worlds w in the structure $\mathcal{K}(t)$ are thus the possible initial states $s \in b_I$, while the worlds that are accessible from a world s to the agent i are the possible initial states s' that are in $r_i(s,t)$. Finally, the valuation associated to a world s in this structure is the state $v(s,t)$ that deterministically follows from the possible initial state s and the action sequence up to $t-1$. $B(t), s \models \phi$ is defined as true when $\mathcal{K}(t), s \models \phi$ is true.

5.4.3. Agent's View

While in the *off-line setting*, a formula ϕ is regarded as true at time t when $\mathcal{K}(t), s_0 \models \phi$ is true for *all* possible initial states $s_0 \in b_I$, i.e., all worlds in the structure, in the *on-line setting*, truth is defined in relation to the single *actual world*, which corresponds to a true but hidden initial state denoted as s_0^*:

Definition 5.4.1 (On-line Truth)**.** A formula ϕ is true at time t in the on-line setting, written $B(t) \models \phi$, iff $\mathcal{K}(t), s_0^* \models \phi$ where $s_0^* \in b_I$ is the hidden initial state.

This a simple but crucial definition in our formulation. No similar explicit account for truth is required in on-line accounts of partially-observable planning in the single-agent setting where the hidden state s_0^* plays an indirect role only. This is because goals are then objective formula and it is then sufficient to keep track of the set of states that are possible at a given time point; the so-called *belief state* (Bonet and Geffner, 2000).

Agents, however, do not have the information to evaluate *arbitrary formulas* according to Definition 5.4.1 as they do not know the hidden state s_0^*. Yet, each agent i can use this definition to evaluate *formulas* $K_i\phi'$ provided that the set $S_i(t)$ of initial states that are possible to agent i by time t is tracked. This set depends on the actual observations gathered by agent i. Initially $S_i(0) = b_I$ and $S_i(t+1)$ is:

$$S_i(t+1) = \{s' | s' \in S_i(t), B(t), s' \models \psi, \forall \psi \in O_i^+(t)\}$$

where $O_i^+(t)$ stands for the set of observations available to agent i at time t; namely, the formulas ψ (observable) in $O_i(t)$ that have been observed to be *true* at t, and the *negation* of the formulas ψ (observable) in $O_i(t)$ that have been observed to be *false*. Provided with this set of possible initial states, the truth of formulas $K_i\phi$ according to Definition 5.4.1 can be evaluated as follows:

Theorem 2. *$B(t) \models K_i\phi$ iff $\mathcal{K}(t), s_0 \models \phi, \forall s_0 \in S_i(t)$.*

Indeed, for evaluating the formula $K_i\phi$ in s_0^*, the agent does not need to know the hidden state s_0^* but $r_i(s_0^*, t)$; i.e., the set of states that agent i cannot tell apart from s_0^* at time t. Yet this set is precisely $S_i(t)$.

As an illustration, if the problem P involves two agents 1 and 2, two fluents p and q, $I = \{p \equiv q\}$, and π is given by the action $\pi(0) =$ **sense**$[1](p)$ followed by $\pi(1) =$ **sense**$[2](q)$, we get a joint belief $B(t)$ for $t = 2$ that defines a Kripke structure $\mathcal{K}(t)$ where formulas such as $K_1 p \equiv K_2 q$ hold in all the states, and formulas such as $K_1 p$ and $K_2 q$ do not. Yet, if the true hidden state s_0^* is such that p and q are true in s_0^*, formulas such as $K_2 q$ and $K_2 K_1 p$ would be true in $B(t)$ according to Definition 5.4.1 for $t = 2$, and false for $t = 1$.

5.5. Planning

Planning in our setting involves the incremental computation and execution of a sequence of actions that makes the goal true. The algorithm shown in Figure 1 computes such sequences using a replanning method that is similar to those developed for single-agent on-line planning in partial observable settings (Brafman and Shani, 2012b; Bonet and Geffner, 2014b). Initially, a selected planning agent i computes an action sequence π by calling a *classical planner* over a translation $K(P, B(t), S_i(t))$ that expresses a *relaxation* where agent i is allowed to make a guess about the true hidden state s_0^*. This simplification does not make the hidden state known to the planning agent but determines the outcomes of all sensing actions which thus become deterministic. If the planning agent i is "lucky", the execution of the (normalized) action sequence π will not reveal to agent i that the choice is wrong. In such a case, the action sequence can be applied fully, achieving K_iG and hence the goal G. On the other hand, if the execution of π reveals to agent i at time $t' > t$ that s is not the true hidden initial state, then s is removed from $S_i(t')$, and the process repeats with the updated beliefs $B(t')$ and sets $S_i(t')$, possibly with a different planning agent. One agent is selected as the planning agent in each replanning episode. A fixed ordering among the agents is also assumed so that if for the selected planning agent i, the classical problem $K(P, B(t), S_i(t))$ has no solution, the selected planning agent becomes the next agent in the ordering. Notice that an action like **sense**$[j](K_i\phi)$ in a plan computed by agent k represents information sharing when $k = i$ and information request when $k = j$. Similarly, a physical action a planned by agent i and owned by agent j represents a request from i to j to do the action a.

Algorithm 1 Online planning and execution for problem P

Inputs: $B(0)$, $S(0)$, initial planning agent i
$t \leftarrow 0$
Loop: Generate classical problem $K(P, B(t), S_i(t))$
Compute **classical plan** π from $K(P, B(t), S_i(t))$
Normalize π removing auxiliary actions
Execute π incrementally **updating** $B(t)$ and $S_i(t)$ til first t' where K_iG achieved or **inconsistency** detected
Agents j **update** $S_j(t)$ til $t = t'$ with own observations
if K_iG achieved **then**
    exit
else
    $t \leftarrow t'$
    Set new planning agent i
    Go to **Loop**

5.5.1. Properties

Before considering the translation in detail, we present the basic properties which can also be understood as the requirements that the translation must fulfill. The translation introduces auxiliary actions, such as assuming a hidden true state and simulating the passive sensors. For an action sequence π obtained from the translation, the *normalization* of π, denoted as $n(\pi)$, is the same sequence but with the auxiliary actions removed. The notion of *consistency* results from matching the observations assumed by the plan and the actual observations gathered. The former follow from the choice of the hidden state which is captured by an auxiliary action $assume(s)$ that must be unique and appear first in the plan.

Definition 5.5.1 (Consistency)**.** Let π be a prefix of a plan for $P' = K(P, B(t), S_i(t))$. The normalized sequence $n(\pi)$ is *consistent* with the observations iff a) for any formula ϕ rendered *observable* by $n(\pi)$ at time t' from active or passive sensing, $B(t'), s \models \phi$ iff ϕ is observed to be true at time t', and $assume(s)$ is the first action in π, and b) the physical actions a in $n(\pi)$ are all applicable in P (i.e., owners know the preconditions).

The results below assume further that a physical action a owned by agent j that is *not* applicable in the plan computed by agent $i \neq j$ from the translation, is replaced by a communication; namely, the action **sense**$_i(K_j(Pre(a)))$. That is, agent i learns that the action is not applicable.

Theorem 3 (Soundness)**.** *a) If π is plan for $K(P, B(t), S_i(t))$ that is consistent with the observations, the execution of $n(\pi)$ leads to the goal in P. b) Otherwise, if π' is the shortest prefix of π that is inconsistent and π includes the action $assume(s)$, after the execution of $n(\pi')$ in P, $s \notin S_i(t')$ where t' is the resulting time step.*

Theorem 4 (Completeness)**.** *If $s = s_0^* \in S_i(t)$ is the true hidden state in P and there is an action sequence that achieves K_iG for an agent i, then there is a plan π for $K(P, B(t), S_i(t))$ that starts with the action $assume(s)$, and any such plan is consistent.*

These properties of the translation ensure that Algorithm 1 is a sound and complete replanning algorithm for P provided that no execution of P can reach a dead-end, i.e., a situation from which no action sequence can lead to K_iG for any agent i:

Theorem 5 (Goal Achievement)**.** *If the executions in P cannot reach a dead-end, Algorithm 1 will solve P after a number of calls to the classical planner that is bounded by $|b_I| \times |A|^2$, where b_I is the set of initial states in P and A is the set of agents.*

In the worst case, a protocol may have to iterate over all the agents until finding an agent i that can find a plan in the translation for the goal K_iG. The execution of that plan ensures that the goal K_iG is reached or that at least one state s is removed from $S_i(t)$. The number of such removals is bounded by $|b_I| \times |A|$.

5.5.2. Translation into Classical Planning

The language for the translation $P' = K(P, B(t), S_i(t))$ in Algorithm 1 is STRIPS extended with *negation*, *conditional effects*, and *axioms*. The primitive fluents in P' are used to represent the states $v(s,t)$ and the collection of states $r_j(s,t)$ that define the beliefs $B(t)$. For encoding the states $v(s,t)$, P' contains atoms L/s that express that the objective literal L is true in the current state if s is the initial state, while for encoding the sets $r_j(s,t)$, P' contains fluents $D_j(s,s')$ that are true when $s' \notin r_j(s,t)$. P' also features atoms $T(s)$ for representing that s is the *assumed true initial state*, and atoms $D_i(s)$ for representing that $s \notin S_i(t)$. Formulas appearing in action preconditions, goals, and sensing expressions in P are assumed to be all literals or conjunctions of possibly epistemic literals L. A positive epistemic literal is an objective literal preceded by a sequence of epistemic operators possibly separated by negations, like $K_a \neg K_b K_c p$. The axioms in the translation are used to maintain the truth of epistemic literals. We denote the set of objective literals in P as $L_F(P)$, the set of positive epistemic literals in P as $L_K(P)$, and the set of positive epistemic literals L that are suffixes of literals in $L_K(P)$ as $L_X(P)$. The literals ϕ/t in the translation are used to encode the truth of formulas ϕ in the assumed initial state; i.e., ϕ/t iff ϕ/s and $T(s)$. Such formulas ϕ are the ones appearing in sensing and preconditions. The actions in $K(P, B(t), S_i(t))$ comprise the physical actions in P, the auxiliary actions $assume(s)$ for guessing the initial state, the action $\mathcal{E}$ for capturing the effects of passive sensing, and the sensing actions **sense**$[A](\phi)$ in P. The action $assume(s)$ must appear first in any plan for some possible s, excluding all other $assume(s')$ actions from being applied.

Definition 5.5.2. The *classical problem with axioms* $K(P, B(t), S_\alpha(t)) = \langle F', I', O', G', X' \rangle$ where α is the planning agent and $P = \langle A, F, I, O, N, S, G \rangle$ is such that:

- $F' = \{L/s : L \in L_F(P), s \in b_I\} \cup \{T(s) : s \in b_I\} \cup \{D_i(s,s') : i \in A, s, s' \in b_I\} \cup \{D_\alpha(s) : s \in b_I\}$,
- $I' = \{L/s : L \in L_F(P), s \in b'(t), s \models L\} \cup \{D_\alpha(s) : s \in b_I, s \notin S_\alpha(t)\} \cup \{D_i(s,s') : s, s' \in b_I, s \notin r_i(s',t), i \in A\}$
- $G' = \wedge_{s \in b_I}(D_\alpha(s) \vee G/s)$
- Axioms X':
    - K_iL/s iff $\wedge_{s' \in b_I}[L/s' \vee D_i(s,s')]$, $K_iL \in L_X(P)$

- ϕ/t iff $\wedge_{s \in b_I}[\neg T(s) \vee \phi/s]$, ϕ in sensing

- Actions O':
  - **auxiliary actions** $assume(s)$, for $s \in b_I$, with prec. $\neg D_\alpha(s)$ and effect $T(s)$,
  - **physical actions** $a \in O$ owned by j have prec. $K_j(Pre(a))/t$ and effects $\neg K_j(Pre(a))/s \rightarrow D_i(s, s') \wedge D_\alpha(s)$ for $s, s' \in b_I$ and $C/s \rightarrow E/s$ for each $s \in b_I$ and effect $C \rightarrow E$ of a in P
  - **sensing actions sense**$[B](\phi) \in N$ with $\alpha \notin B$ mapped into same action without precs, and effects:
    - $\phi/s \wedge \neg\phi/s' \rightarrow D_i(s, s'), D_i(s', s)$ for s, s' in b_I and $i \in B$,
  - **sensing actions sense**$[B](\phi) \in N$ with $\alpha \in B$ mapped into the same actions, with effects
    - $\phi/s \wedge \neg\phi/s' \rightarrow D_i(s, s'), D_i(s', s)$ for s, s' in b_I and $i \in B$, and
    - $\phi/t \wedge \neg\phi/s \rightarrow D_\alpha(s)$,
    - $\neg\phi/t \wedge \phi/s \rightarrow D_\alpha(s)$, for $s \in b_I$,
  - **auxiliary action** $\mathcal{E}$ with effects
    - $\phi/s \wedge \neg\phi/s' \rightarrow D_i(s, s'), D_i(s', s)$ for each pair of states s, s' in b_I, **psense**$[B](\phi)$ in S, and $i \in B$,
    - $\phi/t \wedge \neg\phi/s' \rightarrow D_\alpha(s')$, if $\alpha \in B$, $s, s' \in b_I$.
    - $\neg\phi/t \wedge \phi/s' \rightarrow D_\alpha(s')$, if $\alpha \in B$, $s, s' \in b_I$.

In the above translation we omit the auxiliary literals used for specifying ordering of actions (forcing as first action one of $assume(s)$, action $\mathcal{E}$ being applied after every other action). Also, while not covered in the above translation, sensing actions can be parallel.

The translation is quadratic in the number of possible initial states $|b_I|$, and hence exponential in the number of atoms in the worst case. The same is true however for sound and complete translations in the single-agent setting (Brafman and Shani, 2012a).

5.5.3. Protocols

The results above make no assumption about which agent is selected as the planning agent for the next episode. Yet this choice can make a significant difference in the type of agent dialogues (information exchanges) that result. We consider four *protocols*.

In *fixed agent,* the initial planning agent remains so throughout the execution until reaching the goal.

In *last-agent,* when the shortest inconsistent plan ends with a sensing action **sense**$[B](K_j\phi)$ or a physical action owned by an agent j different than the planning agent, the control is given to agent j.

Third is the *volunteering* protocol. When the shortest inconsistent plan ends with a sensing action involving agent j (eg **sense**$[i](K_jL)$) and i is the planning agent, j "volunteers" information to i. This is achieved by selecting the most informative sensing action of the form **sense**$[i](K_jL')$ to be applied. As most *informative* we define the action which, when applied, will remove the largest number of states from the set of states R that i may consider possible, according to j. Formally, $R = \{ s \mid s \in r_i(s', t)$ and $s' \in S_j(t)\}$ is the set of states i may consider possible, from the perspective of j. Then, for all possible sensing actions **sense**$[i](K_jL')$, we define $R(K_jL') = \{ s \mid s \in R,\ B(t), s \models K_jL'$ iff $B(t) \models K_jL'\}$, the set of all states in R which agree with the truth value of K_jL'. The action with the smallest $|R(K_jL')|$ is chosen as the most informative. Ties break randomly, and no sensing action will be applied if there is no $|R(K_jL')| < |R|$.

The forth, and last, protocol is the *vol-mutex* protocol. Similarly to the *volunteering* protocol, when the shortest inconsistent plan ends with a sensing action involving agent j (eg **sense**$[i](K_jL)$), and i is the planning agent, j "volunteers" information to i. The difference is that instead of j volunteering the most *informative* information, he will volunteer the information most *relevant* to L. We define this relevance using predefined sets of mutex literals - two literals L and L' are relevant if they belong to the same set of mutexes. If the plan ended with a sensing action **sense**$[i](K_jL)$, where i expected K_jL to be true but he actually sensed that it is false, and there exists a literal L' relevant to L such that j knows L', then the sensing action **sense**$[i](K_jL')$ is applied. If for all L' relevant to L we have that $B(t) \not\models K_jL'$, then a parallel sensing action occurs of the form **sense**$[i](K_jL', ..., K_jL'')$ for $L'..L'' \in M$.

The difference between the *volunteering* and the *vol-mutex* protocol is a subtle one. We can see that in the *volunteering* protocol the agent shares the knowledge which will have *possibly* the biggest impact, yet it is possible that the information is irrelevant to the asking agent. Imagine a problem where two balls are placed in a grid. Ball 1 has 20 possible positions while ball 2 only four, the corners of the grid. Imagine agent j already knows the positions of both balls, and i, who is the planning agent, has as goal to learn only the position of ball 2. Agent i will execute a plan where he assumed the ball is in one specific corner of the grid, ask j if he knows it, expecting a positive answer. If in the hidden, true state the ball is in a different corner, j will reply negatively. The *volunteering* protocol specifies that j will announce the position of ball 1, since that removes most of the states, even though it is irrelevant to i's goal. The *vol-mutex* protocol on the other hand, where there is a set of mutexes which contains the four possible positions of ball 2, specifies that j will share the position of the second ball.

5.6. From plans to dialogues

In this section we show that there is an intuitive mapping between executed actions and natural language, based on the fact that all actions are public, sensing actions can be of epistemic literals etc.

- *Acting:* a physical action a with owner i and preconditions $Pre(a)$ is translated into "i: I apply a".

- *Requesting:* a physical action a with owner j and preconditions $Pre(a)$ is translated into "i: j, apply action a", to which a response will follow: "j: I applied a" or "j: I cannot apply a", depending on whether $K_jPre(a)$. If the action has no preconditions, no response will follow since it is known that the action can be applied.

- *Providing:* a sensing action **sense**$[D](K_iL)$ is translated into "i: I tell all agents in D whether I know L", and if $D = A - \{i\}$ then into "i: I do know L", or "i: I do not know L", depending on the hidden true state.

- *Asking:* a sensing action **sense**$[D](K_jL)$, where $i \notin D$, is translated into "i: j, tell all agents in D whether you know L", to which a response will follow "j: I did tell all agents in D". If $i \in D$, the response depends on the view of the plan we have: if we present it from the point of view of the planning agent, the response will be "j: I do (not) know L", otherwise it would be "j: I did tell all agents in D". If $D = A - \{j\}$, then the the question would be "j: do you know L?", while the response will be "j: Yes, I do know L" or "j: I do not know L".

In certain cases, a more natural mapping is possible. For example, if i applies the action "pick-up-red-ball" of which he is the owner, then we would translate the action to "i: I picked up the red ball". Similarly, if we have a sensing action **sense**$[i](K_jL)$, where L represents the fact that j sees the red ball, it would be translated to "i: j, tell me that you see the red ball".

5.7. Examples and experimental results

We present the dialogue traces for three problems, using various protocols. We obtained the results using the on-line replanning algorithm shown, and the FD planner as the classical planner (Helmert, 2006), over a Linux machine at 2.93GHz with 4GB of RAM. In our implementation, each planning phase is a different call to FD, with the corresponding PDDL files. We present experimental results as tuples $\langle S, T, R\rangle$ next to each problem and protocol used. In these tuples, S stands for the average search time , T is the average total time, and R is the average number of replans. Search (total) time is the average search (total) time for each planning phase, while the average number of replans is taken by running the experiments over each possible initial state as the true initial state. An asterisk '*' next to an action indicates that a replanning phase occurred after the action, and we report when a change of planning agent or a volunteering occurred. Due to space, we collapse actions when the execution is clear. For example, a "j, move right twice. Do you see l?" indicates two consecutive physical actions and a sensing action, all relating to j.

5.7.1. Meeting problem

We have two agents (a, b) and a ring-shaped grid of size six $(p_1, .., p_6)$. Within the grid there are three landmarks (l, q, r), each one positioned in either p_2, p_4 or p_6, and no two landmarks can be in the same position. The agents do not know the actual position of the landmarks. It is known that a is initially positioned in either p_1 or p_2, while b in one of p_2, p_4 and p_6. An agent can see a landmark only if they are in the same position. The goal is for agent a to know that both agents are in p_1.

Each agent has a physical action "move-clockwise" and "move-anticlockwise", three sensors for seeing a landmark, and three actions for communicating if he is in the same position with one of the landmarks. We introduce auxiliary derived atoms $i@L$ with definition $\bigvee_{x\in\{2,4,6\}} i@p_x \wedge L@p_x$, where i the agent, L one of the landmarks. Agents can sense their respective auxiliary derived atoms.

We have in total 4 physical actions: "move-clockwise(i)" with conditional effects $i@p_6 \rightarrow \neg i@p_6 \wedge i@p_1$ and $i@p_x \rightarrow \neg i@p_x \wedge i@p_{x+1}$ for $x \in \{1..5\}$, and "move-anticlockwise(i)" with conditional effects $i@p_1 \rightarrow \neg i@p_1 \wedge i@p_6$ and $i@p_x \rightarrow \neg i@p_x \wedge i@p_{x-1}$ for $x \in \{2..6\}$ and $i \in \{a, b\}$. There are 6 sensors **psense**$[i](i@L)$, for $i \in \{a, b\}$ and 6 sensing actions, **sense**$[a](K_b b@L)$, **sense**$[b](K_a a@L$, for $L \in \{l, r, q\}$, representing what the agent sees in the position he is at and what he communicates. The number of possible initial states are 36: 6 possible states due to the initial unknown positioning of landmarks, 2 possible states due to the uncertainty of a's position, and 3 possible states concerning b's positioning $(6 * 2 * 3)$. Goal $G = K_a a@p_1 \wedge K_a b@p_1$.

The following executions assume a hidden true state where a is positioned at p_1, b is positioned at p_4, and the position of the landmarks is: $r@p_2$, $q@p_4$ and $l@p_6$.

Fixed-agent protocol. **Experiments:** $\langle 0.3s, 1.9s, 2.1\rangle$.

1. **A**: B, do you see l?
2. **B**: No, I do not see l.*
3. **A**: B, do you not see q?
4. **B**: No, I do see q.*
5. **A**: B, move clockwise twice. Do you not see l?
6. **B**: No, I do see l.*
7. **A**: I move anticlockwise. I move clockwise.
8. **A**: B, move clockwise.

In order for a to achieve the goal he needs to learn the position of b in terms of landmarks *and* the position of the landmarks on the grid. After the first two questions, a knows b sees q. He then moves b to a different location and a learns that b sees l. Up to this point, a knows that he is in p_1 since he sees no landmark, has learned that l is clockwise next to q and that b is now at the same position with l. Then, a moves to p_2, and by seeing r learns the actual positions of the landmarks, and, subsequently, the position of b.

Last-agent protocol. **Experiments:** $\langle 0.3s, 1.9s, 3.3\rangle$.

1. **A**: B, do you see l?
2. **B**: No, I do not see l.*
3. **B**: A, do you see q?
4. **A**: No, I do not see q.*
5. **A**: B, do you not see q?
6. **B**: No, I do see q.*
7. **B**: A, do you see r?
8. **A**: No, I do not see r.*
9. **A**: B, move clockwise twice. Do you not see l?
10. **B**: No, I do see l.*
11. **B**: A, do you see l?
12. **A**: No, I do not see l.*
13. **A**: I move anticlockwise. I move clockwise. B, move clockwise.

In the above execution, we see that both agents, when they are the *planning agent*, try first to reduce their uncertainty. We have a constant exchange of information, up to the point where a happens to become the planning agent while he knows the hidden true state. If a was at p_1, his last response would have made his position known to b, as well as the fact that he knows b's position as well, allowing b to achieve the goal $K_b K_a a@p_1 \wedge K_b K_a b@p_1$.

Vol-mutex protocol. **Experiments:** $\langle 0.3s, 1.9s, 1.7\rangle$.

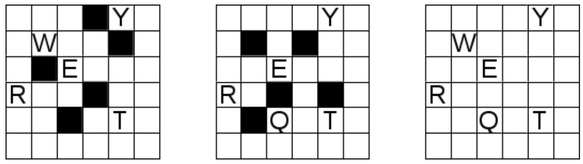

Figure 5.1: Situated dialog example: on the left we see what agent a knows, in the middle what b knows and on the right the hidden, true state. Covered positions on the table indicate the positions that the respective agent cannot see.

1. **A**: B, do you see l?
2. **B**: No, I do not see l.
3. **B**: I do see q.* (*volunteering*)
4. **A**: B, move clockwise twice. Do you not see l?
5. **B**: No, I do see l.*
6. **A**: I move anticlockwise. I move clockwise. B, move clockwise.

Literals $b@q$, $b@r$ and $b@l$ are mutexes: since b can be in only one position, he can see only one landmark. When he is asked about l and he responds negatively, he volunteers the information of what he actually sees, saving a from asking another question.

A general strategy for solving the problem would be for a to move to a position with a landmark, asking b if he sees the same landmark, and if he does not, move him to another position and ask him again the same question. Such a policy is good since it takes into account the issue that replans may be needed. Though such a plan is possible to be found by our approach, the fact that a state is assumed as true in every planning phase leads to optimistic plans, in terms of that assumption. Imagine a assuming a state where he is right next to landmark l and b is in the same position as l. From a's point of view, the plan where he asks first b if he sees l and then a moves left and sees l himself, is the same as first moving to l and then asking b. Yet, the second plan is better considering the possibility of replanning since he at least knows the position of one landmark, while in the first he only learns where b isn't.

5.7.2. Situated dialogue

In this problem, we have a table of size 6x6 (with the $(0, 0)$ coordinates on the top left), six objects (Q, W, E, R, T, Y) placed on it in different positions, and two agents a and b. Each agent can see only part of the table: a can see the entire table *except* of *five* positions which are hidden to him, and it is known that object Q is placed in one of these positions. Similarly, there are five, *different that a's*, positions which are hidden for agent b, and it is known that object W is in one of them. In other words, it is known that each agent can see 5 out of 6 objects and the the positions of four of them

(E, R, T, Y) are known to both, leading to 25 possible initial states (5 for the position of W and 5 for the position of Q).

Objects E, R, T and Y can be moved in four directions by b, and agents can communicate only spatial relationships: they cannot communicate the position of the objects W and Q but they can communicate whether that object is on left/right/over/under another object. The goal is for a to know the position of Q and for agent b to know the position of W.

We have four physical actions for each of the four objects, "move-object-X-right / left / up / down", each with conditionals effect $X@p_{x,y} \rightarrow \neg X@p_{x,y} \wedge X@p_{x',y'}$, for $X \in \{E, R, T, Y\}$, and for all positions $p_{x,y}$ to a new position $p_{x',y'}$, depending on which direction the object is move.

Agent a can communicate whether object W is on the left/right/over/under of one of the E, R, T, and Y, and agent b similarly, for object Q. This means that we have in total 32 sensing actions: 16 sensing actions of a communicating **sense**$[b](K_a WrZ)$, and 16 sensing actions for b: **sense**$[b](K_b QrZ)$, both with $r \in \{right, left, over, under\}$ and $Z \in \{E, R, T, Y\}$. Literals XrZ are derived literals with definitions indicating whether the spatial relationship between X and Z holds. As an example, $WoverE$ is derived by a DNF formula with terms $W@p_{x,y} \wedge E@p_{x,y+1}$, where x, y are the five possible initial positions of W (since W cannot be moved there is no reason to define the derived literal over all possible positions of the grid).

The goal is written as $G = (K_a Q@p_{0,3} \vee K_a Q@p_{1,4} \vee K_a Q@p_{3,3} \vee K_a Q@p_{2,1} \vee K_a Q@p_{4,2}) \wedge (K_b W@p_{1,1} \vee K_b W@p_{1,3} \vee K_b W@p_{3,2} \vee K_b W@p_{3,4} \vee K_b W@p_{4,1})$.

The true hidden initial state for the executions below is shown in Figure 5.1.

Fixed agent protocol; **Experiments:** $\langle 0.92s, 2.86s, 2.1 \rangle$.

> **A**: B, do you know Q is left of E?
> **B**: No, I do not know.*
> **A**: B, do you know Q is left of Y?
> **B**: No, I do not know.*
> **A**: B, do you know Q is under of Y?
> **B**: No, I do not know.*
> **A**: B, move E down. Do you know Q is under E?
> **B**: Yes, I do know.
> **A**: B, move R right. Move R up. I do know W over R.

Initially, agent a tries to find out the position of Q, learning that Q does *not* have a spatial relationship with any object. Object E is then moved next to the remaining possible positions of Q, creating the necessary relationships. When a learns the position of Q, object R is moved to allow a to communicate his knowledge of the position of W.

Volunteering protocol. **Experiments:** $\langle 1.1, 2.9, 1.4 \rangle$.

> **A**: B, do you know Q is left of E?
> **B**: No, I do not know Q is left of E.
> **B**: I do not know Q under Y.* (*volunteering*)
> **A**: B, do you know Q is left of Y?
> **B**: No, I do not know Q is left of Y.*
> **A**: B, move E down. B, do you know Q is under E?
> **B**: Yes, I do know Q is under E.
> **A**: B, move R right. B, move R up. I do know W is over R.

After the first question of a, to which b replies negatively, b volunteers that he *also* does not know that Q is under Y. If Q *had* a spatial relationship with another object, b would choose to volunteer that relationship, after which a would know the hidden true state and with one planning phase achieve the goal. Since there is no such relationship, volunteering that Q does *not* have a spatial relationship with an object removes 5 states from the set of possible initial states.

5.7.3. The Lights problem

In this problem there are four lights (l_1, l_2, l_3, l_4) and three agents (a, b, c). Initially it is known that at least one of the lights is on. No agent can see the lights themselves, but agent b can sense whether at least one of the lights l_1 and l_2 are on ($l_1 \vee l_2$). Similarly, agent c can sense ($l_3 \vee l_4$), while a cannot sense anything about the physical world. Additional to these two passive sensors, there are eleven sensing actions. In one sensing action agent b is the sensing agent, **sense**$[b](K_c(l_3 \vee l_4))$, while in the other ten it is—— agent a: **sense**$[a](K_b(l_1))$, **sense**$[a](K_b(l_2))$, **sense**$[a](K_b K_c L))$, **sense**$[a](K_b \neg K_c L)$, **sense**$[a](K_b K_c \neg L)$, and **sense**$[a](K_b \neg K_c \neg L)$, with $L \in \{l_3, l_4\}$. Simply, c can communicate his knowledge about what he senses only to b, and b can communicate to a his knowledge about the lights he can sense and his knowledge about the knowledge of c concerning l_3 and l_4. Lastly, there are four physical actions "toggle(L)", for $L \in \{l_1, l_2, l_3, l_4\}$, that toggle light L: turn it on if it was off, and off if it was on, whose owner is a. The goal is for a to know that all lights are on: $K_a l_1 \wedge K_a l_2 \wedge K_a l_3 \wedge K_a l_4$.

The true hidden initial state for the execution is the one where only l_2 and l_4 are on. Note that when b responds to a, c is only aware that b responded and *not* the actual response. Similarly, when c responds to b, a is only aware that a communication took place and not what it was shared. In the execution we show the actual response of b since a is the planning agent.

Fixed agent protocol; **Experiments:** $\langle 0.8s, 1.5s, 3.2\rangle$.

1. **A**: B, tell *me*, do you know that C does not know that l_4 is off?
2. **B**: No, I do not know.*
3. **A**: C, tell B whether you know $l_3 \vee l_4$.
4. **C**: I told B.
5. **A**: B, tell *me*, do you know that C knows that l_4 is off?
6. **B**: No, I do not know that.*
7. **A**: I toggle the second light. B, tell *me*, do you know l_1 is on?
8. **B**: No, I do not know it.*
9. **A**: I toggle the first, the second and the third light. C, tell B whether you know $l_3 \vee l_4$.
10. **C**: I told B.
11. **A**: B, tell *me*, do you not know that C knows that l_4 is on?
12. **B**: Yes, I do.
13. **A**: I toggle the fourth light. C, tell B whether you know $l_3 \vee l_4$.
14. **C**: I told B.
15. **A**: B, tell *me*, do you know that C knows that l_3 is on?
16. **B**: Yes, I do know that.
17. **A**: I toggle the fourth light.

Agent b's first response allows a to derive that $l_1 \vee l_2$ is true. Otherwise, b would know $l_3 \vee l_4$ is true (at least one of l_k must be initially on) and since c can sense $l_3 \vee l_4$, b would also know that c could not know l_4 was off. After c tells b what he sensed (4), b knows that c knows either both l_3 and l_4 to be off, or that at least one of them is on. Since b does not know that c knows l_4 is off, a is able to derive that $l_3 \vee l_4$ is true. Toggling l_2 at step 7, while $l_1 \vee l_2$ is true, creates a situation where either both are off or l_1 is definitely on. The response of b allows a to derive both are off, and turning them on at step 9. Similarly, for achieving $l_3 \wedge l_4$.

5.8. Related Work

In recent years, there has been a growing interest in multi-agent epistemic planning with a number of works placing emphasis on different aspects of the problem. Some place the focus on expressivity and modeling (Baral et al., 2012; Cooper et al., 2016), others in distributed computation and coordination (Engesser et al., 2015), while the most closely related approaches focus on computational issues and the use of classical planners (Brenner, 2010; Brafman et al., 2013; Muise et al., 2015). The works most relevant to ours are (Muise et al., 2015) and (Cooper et al., 2016). A key difference to our approach is they can only represent beliefs about literals, not about arbitrary formulas. This is how they manage to reason about nested beliefs without using explicit or implicit Kripke structures.

5.9. Conclusion

We have extended the belief tracking formulation, presented in the previous chapter, to the on-line setting where plans are supposed to work for the true hidden state as revealed by the observations, and have developed an alternative translation into classical planning for selecting actions within a replanning architecture. Planning is done from the perspective of the agents themselves that have beliefs about the world and nested beliefs about each other. As in the single-agent setting, the replanning approach ensures that goals are reached in a bounded number of episodes provided that dead-ends are not reached.

We have shown that interesting agent dialogues can arise in the proposed setting where agents collaborate by requesting or volunteering information in a goal-directed manner. In spite of the restrictions, however, the approach is not yet scalable, as only problems with tens of possible initial states can be handled in this way. One way for scaling up further to have a more practical dialog system is by adapting the techniques that have been used to improve scalability in the single-agent setting.

PART III

Optimizations and Variations

CHAPTER **6**

A Linear Translation

The previous translations we presented in Chapters 4 and 5 are quadratic in the number of states due to the use of the $D_i(s, s')$ literals for representing the accessibility relations between two states given an agent. In this chapter we will present a translation which is linear in the number of states, and can be used for problems where sensing actions and passive sensors which are not common to all agents involve only static literals. A literal is static if there is no action or axiom in the problem that changes the initial truth value of the literal in a state. We take advantage of this property in a preprocessing phase in order to identify which sensing actions allow the agents to distinguish between which states.

6.1. Requirements for a Linear Translation

The formulations we have seen in the previous chapters (for linear, multi-agent planning problems and for on-line multi-agent problems) share the same requirements:

1. Agents share a common, initial belief on the set of initially possible states.
2. Actions are deterministic.
3. The sequence of events that can change the physical world or the knowledge of the agents is public.

For each of the formulations, we also provided a translation to classical planning, which allowed us to use off-the-shelf planners. The translations are quadratic in the number of initial states due to the sets $r_i(s,t) \in B(s,t)$, for all $s \in b_I$, denoting the accessibility relations between states in the corresponding Kripke structure.

The linear translation we present here introduces a fourth requirement:

All literals that appear in a sensing action or a sensor are static.

By static, we refer to objective literals whose truth value cannot be changed during planning. Epistemic literals are considered as non-static, even when they involve objective literals which are static. The reason is that even though the truth value of an objective literal cannot change, an agent's knowledge of its truth value can (assuming that initially all agents consider all states as possible and the truth value of the static literal is not common knowledge).

This fourth requirement allows us to predefine which states an agent can distinguish from a specific state, based on the sensing actions/sensors that have occurred so far, instead of keeping track of the accessibility relations through a quadratic number of literals. In the following examples, where we assume some familiarity with the previous translations, we show the intuition of how we take advantage of the fourth requirement.

Example 6.1.1. *Suppose a problem P with three states ($s_1 = \{p, q\}, s_2 = \{p, \neg q\}$ and $s_3 = \{\neg p, q\}$) and only one sensing action $sense[1](p)$. In the translations we have presented to classical planning, the conditional effects of the action would be written (simplistically):*

$$
\begin{aligned}
p/s_1 \wedge \neg p/s_2 &\rightarrow D_1(s_1, s_2)\\
\neg p/s_1 \wedge p/s_2 &\rightarrow D_1(s_1, s_2)\\
p/s_1 \wedge \neg p/s_3 &\rightarrow D_1(s_1, s_3)\\
\neg p/s_1 \wedge p/s_3 &\rightarrow D_1(s_1, s_3)\\
p/s_2 \wedge \neg p/s_3 &\rightarrow D_1(s_2, s_3)\\
\neg p/s_2 \wedge p/s_3 &\rightarrow D_1(s_2, s_3)
\end{aligned}
$$

Then $D_1(s, s')$ literals are used in the definition of derived, epistemic literals. We remind the reader that the definition of an axiom for $K_i L/s$ is conditioned on the accessibility relations that s has with the rest of the states. For example, the definition of the axiom for $K_1 q/s_2$ would be:

$$(q/s_1 \vee D_1(s_1, s_2)) \wedge (q/s_2 \vee D_1(s_2, s_2)) \wedge (q/s_3 \vee D_1(s_2, s_3))$$

Suppose now that p *is static. We can take advantage of this fact by having the effects of the sensing action represent only that the action has occurred. The effect of* $sense[1](p)$ *would then be just an auxiliary literal:*

$$(1\_sensed\_p)$$

Since we already know the states that disagree in the truth value of p*, the definition of the axiom for* $K_1 q/s_2$ *can be written as:*

$$q/s_1 \wedge (q/s_2 \vee D_1(s_2, s_2)) \wedge (q/s_3 \vee (1\_sensed\_p))$$

where q/s_1 *must be true since there is no sensing action that would allow agent* 1 *to dinstinguish between* s_1 *and* s_2*, while* $D_1(s_2, s_3)$ *has been replaced by* (1_sensed_p)*. The literal* $D_1(s_2, s_2)$ *denotes that* s_2 *is not considered possible by agent* 1*.*

In our previous formulations, sensing actions are used to depict not only the agent's observations of the environment, but also communication between agents: sensing the epistemic literal $K_i L$ represents the fact that agent i is communicating if he knows L or not.

The fourth requirement, necessary for our linear translation, is quite restrictive by excluding sensing of epistemic literals. For this reason, we introduce a modification of the on-line belief representation of Chapter 5 which allows us to explicitly *remove* states from the set of possible states. In the on-line translation of Chapter 5, such removal of states was implicit: given a sensing action where all agents sensed a formula ϕ at the same time, all accessibility relations leading to states which disagreed with the assumed state on the truth value of ϕ were removed. The same states where also removed from the set of states the planning agent i considered possible ($S_i(t)$) based on his *actual* observations. Thus, those states were rendered irrelevant in the valuation of epistemic literals. We can achieve the same effect by removing the states themselves, which allows us to drop the explicit representation of the accessibility relation between pair of states.

Even though such a modification does not allow us to have an action $sense[B](\phi)$ where $B \subset A$ and ϕ contains atoms which are not static, it allows for sensing actions of the form $sense[A](\phi)$, where A is the set of agents. Specifically, applying a sensing action **sense**$[B](\phi)$, or a passive sensor **psense**$[B](\phi)$, $B \subset A$, has the same effects as in Section 5.4.1, while the action **sense**$[A](\phi)$ updates the sets $r_i(s, t+1)$:

$$r_i(s, t+1) = \{s' \mid s' \in r_i(s, t) \text{ and } B(t), s' \models \phi\} \tag{6.1}$$

for $i \in A$ and $s \in b_I$.

We can see the intuition of this approach in the following example.

Example 6.1.2. *We extend Example 6.1.1 with (i) an assumed hidden, true state s_t representing either s_1, s_2 or s_3, (ii) the auxiliary literals $D(s_i)$ denoting that state s_i is not considered possible by the agents, and (iii) with the sensing action $sense[A](K_1 q)$, where q is not static and A is the set of agents.*

The definition of the axiom for $K_1 q/s_2$ becomes:

$$(q/s_1 \vee D(s_1)) \wedge (q/s_2 \vee D(s_2)) \wedge (q/s_3 \vee D(s_3) \vee (1\_sensed\_p))$$

which, after the action $sense[1](p)$ is applied, is false since both q/s_2 and $D(s_2)$ are false. Again, there is no sensing action which can allow agent 1 to distinguish between states s_1 and s_2. Thus, the clause $(q/s_1 \vee D(s_1))$ does not contain any auxiliary literal, while the clause $(q/s_3 \vee D(s_3) \vee (1\_sensed\_p))$ does.

The same occurs with $K_1 q/s_1$, which happens to have the same definition:

$$(q/s_1 \vee D(s_1)) \wedge (q/s_2 \vee D(s_2)) \wedge (q/s_3 \vee D(s_3) \vee (1\_sensed\_p))$$

and it is false for the same reasons. On the other hand, the axiom deriving $K_1 q/s_3$ with the definition:

$$(q/s_1 \vee D(s_1) \vee (1\_sense\_p)) \wedge (q/s_2 \vee D(s_2) \vee (1\_sense\_p)) \wedge (q/s_3 \vee D(s_3))$$

is true, since (1_sense_p) is true, which makes the first and second clauses true, and q/s_3 is true, which, ultimately, makes the definition true.

The effects of $sense[A](K_1 q)$ is now written as:

$$\begin{array}{l}
K_1 q/s_t \wedge \neg K_1 q/s_1 \rightarrow D(s_1) \\
K_1 q/s_t \wedge \neg K_1 q/s_2 \rightarrow D(s_2) \\
K_1 q/s_t \wedge \neg K_1 q/s_3 \rightarrow D(s_3) \\
\neg K_1 q/s_t \wedge K_1 q/s_1 \rightarrow D(s_1) \\
\neg K_1 q/s_t \wedge K_1 q/s_2 \rightarrow D(s_2) \\
\neg K_1 q/s_t \wedge K_1 q/s_3 \rightarrow D(s_3)
\end{array}$$

where the number of conditional effects is twice the number of initial states ($2 * |b_I|$). If $s_t = s_3$, then $D(s_1)$ and $D(s_2)$ become true (denoting the two states as impossible), as expected since s_3 is the only state where $K_1 q$ is true. If $s_t \neq s_3$, only $D(s_3)$ becomes true.

The fourth requirement now becomes less restrictive, and can be written as:

All literals that appear in sensors and sensing actions, that do not involve all agents, must be static.

6.2. Preprocessing

Before defining the linear, classical translation of an on-line multi-agent planning problem P, we need to identify all pairs of states that are distinguishable from the point of view of an agent, based on his sensing actions and passive sensors.

We consider planning problems $P = \langle A, F, I, O, N, S, G \rangle$ where A is the set of agent names or indexes, F is the set of relevant atoms or fluents, I represents the initial situation in the form of an objective formula over F, O is the set of *physical actions*, N is the set of *sensing actions*, S is the set of (passive) *sensors*, and G is the goal. States represent truth-valuations over F, and the set of possible initial states b_I is made of the states that satisfy I. A sensing action in N is a set of expressions of the form **sense**$[i](\phi)$, where i is an agent, and ϕ is an objective formula. A (parallel) sensing action in N is a set of expressions of the form **sense**$[A_k](\phi)$, where the truth of ϕ is revealed to all the agents $j \in A_k$. We denote passive sensors like sensing actions but with the letter "p" in front; namely, as **psense**$[i](\phi)$ and **psense**$[A_k](\phi)$, where $A_k \subset A$. For presenting the precompilation, we will refer to both **sense**$[A_k](\phi)$ and **psense**$[A_k](\phi)$ as $\alpha[A_k](\phi)$.

We will assume that the preprocessing is applied to problems for which we know that all four requirements mentioned in the previous section are satisfied. This means that for every sensing action $sense[A_k](\phi)$ or passive sensor $psense[A_k](\phi)$, where $A_k \subset A$, ϕ must contain only static literals. Thus, checking whether sensing actions/passive sensors involve only static literals, or they involve all agents, is not part of the preprocessing.

We define O^+ as a vector denoting the distinguishability between pairs of states for a specific agent:

$$O^+ = \{ \ O^+(s, s', i) \mid s, s' \in b_I, i \in A \ \} \tag{6.2}$$

where $O^+(s, s', i)$ is a set of sensing actions and passive sensors, where:

$$O^+(s, s', i) = \{ \ \alpha[A_k](\phi) \mid \alpha[A_k](\phi) \in N \cup S, i \in A_k, A_k \subseteq A, s \models \phi, s' \not\models \phi \ \} \tag{6.3}$$

In other words, a sensing action or a passive sensor belongs in the set $O^+(s, s', i)$ if by applying the sensing action or the passive sensor, agent i gets to distinguish between states s and s'.

6.3. Linear Translation into Classical Planning

The language for the linear translation $P' = K(P, B(t), S_i(t), O^+, O(t))$ is STRIPS extended with *negation*, *conditional effects*, and *axioms*. The addition to P' is the set O^+, which was computed during the preprocessing, and the set $O(t)$ which is the set of sensing actions/passive sensors that have been applied up to time t.

For encoding the states $v(s,t)$, P' contains atoms L/s that express that the objective literal L is true in the current state, if s is the true, initial state. P' also features atoms $T(s)$ for representing that s is the *assumed true initial state*, and atoms $D_i(s)$ for representing that $s \notin S_i(t)$, while atoms $D(s)$ are used for representing that state s is no longer possible and this fact is common knowledge among the agents. We will use $s \notin Poss$ to denote that s is no longer a possible initial state. In other words, if $s \notin Poss$, then $s \notin r_i(s', t)$, for $i \in A$ and $s' \in b_I$.

Additionally, for each sensing action and passive sensor $\alpha[A_k](\phi) \in N \cup S$, $A_k \subset A$ of P, we introduce an atom $a^+_{A_k,\phi}$ denoting that the sensing action/passive sensor has been applied. These atoms belong to $O(t)$ if the sensing action/passive sensor they correspond to has been applied during execution of the plan.

Formulas appearing in action preconditions, goals, and sensing expressions in P are assumed to be all literals or conjunctions of possibly epistemic literals L. A positive epistemic literal is an objective literal preceded by a sequence of epistemic operators possibly separated by negations, like $K_a \neg K_b K_c p$.

The axioms in the translation are used to maintain the truth of epistemic literals. We denote the set of objective literals in P as $L_F(P)$, the set of positive epistemic literals in P as $L_K(P)$, and the set of positive epistemic literals L that are suffixes of literals in $L_K(P)$ as $L_X(P)$.

The literals ϕ/t in the translation are used to encode the truth of formulas ϕ in the assumed initial state; i.e., ϕ/t iff ϕ/s and $T(s)$. The actions in $K(P, B(t), S_i(t), O^+, O(t))$ comprise the physical actions in P, the auxiliary actions $assume(s)$ for guessing the initial state, the action $\mathcal{E}$ for capturing the effects of passive sensing, and the sensing actions **sense**$[A](\phi)$ in P. The action $assume(s)$ must appear first in any plan for some possible s, excluding all other $assume(s')$ actions from being applied.

Definition 6.3.1. The linear, *classical problem with axioms* $K(P, B(t), S_\alpha(t).O^+, O(t)) = \langle F', I', O', G', X' \rangle$ where α is the planning agent and $P = \langle A, F, I, O, N, S, G \rangle$ is such that:

- $F' = \{L/s : L \in L_F(P), s \in b_I\} \cup \{T(s) : s \in b_I\} \cup \{D(s) : s \in b_I\} \cup \{D_\alpha(s) : s \in b_I\} \cup \{a^+_{A_k,\phi} : a[A_k](\phi) \in N \cup S\}$,

- $I' = \{L/s : L \in L_F(P), s \in b'(t), s \models L\} \cup \{D_\alpha(s) : s \in b_I, s \notin S_\alpha(t)\} \cup \{D(s) : s \notin Poss\}, \cup \{a^+_{A_k,\phi} : a[A_k](\phi) \in O(t)\}$
- $G' = \wedge_{s \in b_I}(D_\alpha(s) \vee G/s)$
- Axioms X':
  - K_iL/s iff $\wedge_{s' \in b_I}[L/s' \vee D(s') \bigvee_{a[A_k])(\phi) \in O^+(s,s',i)} a^+_{A_k,\phi}]$, $K_iL \in L_X(P) \cup L_K(P)$, $O^+(s,s',i) \in O^+$,
  - ϕ/t iff $\wedge_{s \in b_I}[\neg T(s) \vee \phi/s]$,
- Actions O':
  - **auxiliary actions** $assume(s)$, for $s \in b_I$, with prec. $\neg D_\alpha(s)$ and effect $T(s)$,
  - **physical actions** $a \in O$ owned by j have prec. $K_j(Pre(a))/t$ and effects $\neg K_j(Pre(a))/s \rightarrow D(s) \wedge D_\alpha(s)$ for $s' \in b_I$ and $C/s \rightarrow E/s$ for each $s \in b_I$ and effect $C \rightarrow E$ of a in P
  - **sensing actions sense**$[B](\phi) \in N$ with $\alpha \notin B$ mapped into same actions with effect:
    - $a^+_{B,\phi}$,
  - **sensing actions sense**$[B](\phi) \in N$ with $\alpha \in B$ mapped into the same actions, with effects
    - $a^+_{B,\phi}$, and
    - $\phi/t \wedge \neg\phi/s \rightarrow D_\alpha(s)$,
    - $\neg\phi/t \wedge \phi/s \rightarrow D_\alpha(s)$, for $s \in b_I$,
  - **sensing actions sense**$[B](\phi) \in N$ with $B = A$ mapped into the same actions, with effects
    - $\phi/t \wedge \neg\phi/s \rightarrow D(s) \wedge D_\alpha(s)$,
    - $\neg\phi/t \wedge \phi/s \rightarrow D(s) \wedge D_\alpha(s)$, for $s \in b_I$,
  - **auxiliary action** $\mathcal{E}$ with effects
    - $a^+_{B,\phi}$ for each **psense**$[B](\phi)$ in S, and
    - $\phi/t \wedge \neg\phi/s \rightarrow D_\alpha(s)$,
    - $\neg\phi/t \wedge \phi/s \rightarrow D_\alpha(s)$, if $\alpha \in B$, $s \in b_I$.

In the above translation we omit the auxiliary literals that are used for specifying ordering of the actions (forcing as first action one of $assume(s)$ and that no other action $assume(s')$ can be applied, action $\mathcal{E}$ being applied after every action).

The translation is linear to the number of initial states, and it has the same properties as the quadratic translation, presented in Section 5.5.1. We can replace $K(P, B(t), S_i(t))$ in Algorithm 1 with the linear translation $K(P, B(t), S_\alpha(t).O^+, O(t))$, with the additional requirement that while executing plan π we also update $O(t)$ by adding to it all sensing actions that have been applied.

Theorem 6 (Soundness). *a) If π is plan for $K(P, B(t), S_i(t), O^+, O(t))$ that is consistent with the observations, the execution of $n(\pi)$ leads to the goal in the problem P. b) Otherwise, if π' is the shortest prefix of π that is inconsistent and π includes the action $assume(s)$, after the execution of $n(\pi')$ in P, $s \notin S_i(t')$ where t' is the resulting time step.*

Theorem 7 (Completeness). *If $s = s_0^* \in S_i(t)$ is the true hidden state in P and there is an action sequence that achieves K_iG for an agent i, then there is a plan π for $K(P, B(t), S_i(t), O^+, O(t))$ that starts with the action $assume(s)$, and any such plan is consistent.*

A difference between the linear translation and the quadratic translation of Section 5.5.2 is the size of the axioms used to derive epistemic literals. In the quadratic translation, the definition of the axioms are 2-CNF formulas. Specifically, each of the axioms has a definition of n clauses (one clause for each possible, initial state), where each clause has a size of 2 ($L/s \vee D_i(s, s')$). In the linear translation, the number of the clauses is the same but their size may differ, since they depend on the number of sensing actions that allow two states s and s' to be distinguished.

CHAPTER **7**

A Decomposition Approach

Decomposition approaches have been used in planning in order to obtain more compact representations of the original problem (Bonet and Geffner, 2014a). The intuition behind decomposition approaches is, instead of tackling the problem in its entirety, to divide it into subproblems, and take into consideration at each step only the literals which are relevant to each subproblem.

Suppose a problem where the initial position of two agents, 1 and 2, on a 10×10 grid is known to both agents. Suppose also that there are two balls, b_1 and b_2, on the grid, where each one can be in any position. Agents can see a ball only if they are in the same position, and the goal is for agent 1 to know the position of b_1 and for agent 2 to know the position of b_2. This problem has 10000 initial states: 100 possible positions for the first ball, and 100 for the second. Intuitively, we can see that the problem of finding one ball (and communicating its position) is independent of the problem of finding the second. The knowledge an agent might have about the position of one of them does not provide any information about the position of the other one. The idea behind the decomposition approach is, rather than solving one problem with 10000 states, to have a formulation that allows us to model and solve two problems of 100 states each. [1].

In this section, we present a simple decomposition approach which addresses the size of the joint belief.

[1] In case there was an initial restriction such as ”the two balls cannot be in the same position”, then the position of one ball does provide information for the position of the other. Mainly, if we know a ball is in a certain position, we know that the other ball cannot be in the same position.

7.1. Decomposition through Relevance

We consider planning problems $P = \langle A, F, I, O, N, S, G\rangle$ where A is the set of agent names or indexes, F is the set of relevant atoms or fluents, I represents the initial situation in the form of an objective formula over F, O is the set of *physical actions*, N is the set of *sensing actions*, S is the set of (passive) *sensors*, and G is the goal. We refer the reader to Chapter 5 for further details.

7.1.1. Initial Decomposition

Before we define the *decomposed* belief representation, we need to specify how we compute the decompositions, given a planning problem P. This is achieved by considering two notions of relevance between literals: the *initial* relevance, and the *dynamic* relevance.

The initial relevance between two literals is defined over the initial formula I. Intuitively, given an agent i, two literals L and L' are initially relevant if agent i can derive the truth value of L' by learning the truth value of L. As an example, consider a problem P' with $I' = (p \vee q)$ and $A = \{i\}$. If agent i knows that p is false, he is able to derive that q must be true. In this example, literals p and q are initially relevant.

In other words, the initial decomposition captures the fact that observations of certain literals allow an agent to infer the truth value of other literals.

Given the formula I denoting the initial situation for the problem P, the question is how do we derive all pairs of initially relevant literals. To achieve it, we will use the notion of *prime implicates* (Darwiche and Marquis, 2002):

Definition 7.1.1. A clause λ is a prime implicate of a formula I if and only if:

- $I \models \lambda$, and
- if $I \models \lambda'$ such that $\lambda' \models \lambda$, then $\lambda \models \lambda'$.

We will write $PI(I)$ to denote the set of prime implicates of formula I. In other words, $PI(I)$ is a set of clauses such that every clause which can be derived from I is subsumed by a clause in $PI(I)$, and no clause in $PI(I)$ can be subsumed by another clause in $PI(I)$.

Given that the set $PI(I)$ is the set of the strongest, derivable clauses from the initial situation denoted by I, we can now define the initial relevance as:

Definition 7.1.2. Given a planning problem $P = \langle A, F, I, O, N, S, G \rangle$, and the set of clauses $I' = PI(I) \cup (l \vee \neg l)$, for every $l \in F$ and $l \notin \mathcal{L}(PI(I))$, two literals $L, L' \in F$ are initially relevant if and only if:

1. there exists a clause $C \in PI(I)$ such that $L, L' \in C$, or
2. there exists a literal L'' such that L is initially relevant to L'' and L'' is initially relevant to L'.

The initial relevance of literals L that appear in F defines a partition, where each subset contains only literals which are relevant to each other. We call this partition the initial decomposition $Q(P)$:

$$Q(P) = \{Q_1(P), ..., Q_n(P)\}$$

where each $Q_j(P) \in Q(P)$ denotes a different subset of initially relevant literals.

Definition 7.1.3. Given a planning problem P with I the formula depicting the initial situation and O the set of physical actions, a literal L is *common*, denoted as $c(L)$, if L appears as a unit clause in the prime implicates of I, and for each conditional effect $e : C \to E$ of the actions in O, if L appears in the effects E then C contains only *common* literals.

In other words, a *common* literal is a literal whose truth value is guaranteed to be always known to the agents. Common literals can only appear as unit clauses in the prime implicates of the initial formula, which means they appear alone in their corresponding set $Q_j(P)$. We define

$$Q_c = \{\ Q_j(P) \mid Q_j(P) \in Q(P), Q_j(P) = \{L\}, c(L)\}$$

as the set of all common literals and we rewrite the initial decomposition $Q(P)$ as:

$$Q'(P) = \{Q'_1(P), ..., Q'_n(P)\}$$

where $Q'_j(P) = Q_j(P) \cup Q_c$.

The following definition will help us define the dynamic relevance.

Definition 7.1.4. Given a formula ϕ with objective and epistemic literals, $\mathcal{L}(\phi)$ is the set of all objective literals that appear in ϕ, either by themselves or as part of epistemic literals.

As an example, given $\phi = (\neg p \wedge K_i(q \rightarrow K_j \neg r))$, we have $\mathcal{L}(\phi) = \{p, q, r\}$.

7.1.2. Dynamic Decomposition

We define the dynamic decomposition of a problem P through the initial decomposition $Q(P)$:

Definition 7.1.5. Given a planning problem P and a partial decomposition $Q(P)$, two sets $Q_j(P), Q_k(P) \in Q(P)$ are dynamically relevant if and only if:

1. there exists an action $\alpha \in O$ with a conditional effect $e : C \rightarrow E$ such that there are at least two, not common literals L and L' for which we have $L, L' \in \mathcal{L}(C) \cup \mathcal{L}(E)$, $L \in Q_j(P)$ and $L' \in Q_k(P)$.
2. there exists a sensing action $o \in N$ or a sensor $o \in S$, which denotes some agent(s) sensing the truth value of a formula ϕ, such that there are at least two, not common literals L and L' for which we have $L, L' \in \mathcal{L}(\phi)$, $L \in Q_j(P)$ and $L' \in Q_k(P)$.
3. there exists a sensing action or a physical action with precondition Pre such that $L, L' \in \mathcal{L}(Pre)$, $L \in Q_j(P)$ and $L' \in Q_k(P)$.
4. there exists a clause C in goal G such that there are at least two, not common, literals L and L' for which we have $L, L' \in \mathcal{L}(C)$, $L \in Q_j(P)$ and $L' \in Q_k(P)$.
5. there exists a set $Q_l(P)$ such that $Q_j(P)$ is dynamically relevant to $Q_l(P)$ and $Q_l(P)$ is dynamically relevant to $Q_k(P)$.

While the initial relevance defines a partition $Q(P)$ of F, the dynamic relevance defines a partition $D(P)$ of $Q(P)$. We call $D(P)$ a decomposition of P.

$$D(P) = \{D_1(P), ..., D_n(P)\}$$

Each $D_k(P)$ is a union of dynamically relevant sets from $Q(P)$. Lastly, for every $D_k(P) \in D(P)$ we define $I(D_k)$ to be:

$$I(D_k(P)) = \{\ C \mid C \in PI(I), \mathcal{L}(C) \subseteq D_k(P)\ \}$$

In other words, given an on-line multi-agent planning problem P, every set of literals $D_k(P) \in D(P)$ defines:

- a subformula $I(D_k(P))$ which contains only the clauses of I whose literals are in $D_k(P)$.
- a set of states $S(D_k(P))$ denoted by the subformula $I(D_k(P))$. States $s \in S(D_k(P))$ will be called *partial* states since they contain only a subset of the set of literals F of the original problem P.

7.2. Belief Representation and Updates

7.2.1. The Decomposed Joint Belief

We define the beliefs of all agents at time t as the decomposed joint belief $B_D(t)$ and, given a decomposition $D(P)$, it is represented by a vector of *partial* beliefs:

$$B_D(t) = \{ \ B_1(t), ..., B_n(t), B_c(t) \ \}$$

Each partial belief $B_j(t) \in B_D(t)$ corresponds to a set $D_j(P) \in D(P)$ and it is a tuple representing the beliefs of the agents conditioned on partial states:

$$B_j(t) = \{ \ B_j(s,t) \mid s \in S(D_j(P)) \ \}$$

with

$$B_j(s,t) = \langle v^j(s,t), r_1^j(s,t), ..., r_n^j(s,t) \rangle$$

where s is a partial state from $S(D_j(P))$, $v^j(s,t)$ the partial state which resulted after executing an event sequence $e(0), ..., e(t-1)$, and $r_i^j(s,t)$ the set of partial, initial states that agent i cannot distinguish from s at time t, for all agents $i \in A$.

For $B_j(s,t) \in B_j(t)$ and $t = 0$ we have that $v(s,t) = s$ and $r_i(s,t) = S(D_j(P))$. In other words, initially, agents cannot distinguish between the states of a partial belief.

7.2.2. Updates

The definition of dynamic relevance ensures that for all the passive sensors, sensing actions and conditional effects of physical actions of a problem P (which we will call *events*), there exists a partial belief $B_j(t)$ whose partial states contain truth values for all the literals in the event.

The effects that each event has on the corresponding partial belief is the same as denoted in the belief representation $B(t)$ of Chapter 5, by extending sensing actions with preconditions and owners as in physical actions.

We can identify the partial belief $B_j(t)$, on which an event $e(t)$ will be applied, by checking whether the literals that appear in $e(t)$, either as part of a formula ϕ or within a conditional effect $C \rightarrow E$, also appear in the truth valuations of the partial states of $B_j(t)$. Specifically:

Definition 7.2.1. Given a decomposed, joint belief $B_D(t)$, based on a decomposition $D(P)$, a partial joint belief $B_j(t) \in B_D(t)$ ***corresponds*** to a formula ϕ if and only if all literals in ϕ appear in the set of dynamically related literals $D_j(P)$.

For each physical action, sensing action and sensor in a problem P, we can identify a partial joint belief to which the update will be applied. Assuming all formulas are in conjunctive normal form, we have:

- given an event $sense[A](\phi)$ or $psensor[A](\phi)$, the event will be applied on the partial belief $B_j(t)$ for which we have that $\mathcal{L}(\phi) \subseteq D_j(P)$,
- given a physical action α, each conditional effect $e : C \rightarrow E$ which contains only common literals will be applied to each partial belief $B_j(t) \in B_D(t)$, while the remaining conditional effects will be applied to the partial belief $B_j(t)$ for which we have $\mathcal{L}(C) \subseteq D_j(P)$ and $\mathcal{L}(E) \subseteq D_j(P)$.
- given any event with preconditions Pre, the precondition will be evaluated on the partial belief $B_j(t)$, for which we have $\mathcal{L}(Pre) \subseteq D_j(P)$.

Due to Definition 7.1.5, it is guaranteed that there exists a unique partial joint belief for each sensing action, sensor and conditional effect of a physical action.

7.3. From $B_D(t)$ to Kripke Structures

We defined the decomposed, joint belief $B_D(t)$ as a vector of partial beliefs. Each partial belief $B_j(t) \in B_D(t)$ corresponds to a Kripke structure $\mathcal{K}_j(t) = \langle W^t, R^t, V^t \rangle$ defined by $B_j(t)$ where

- $W^t = \{s \mid s \in S(D_j(P))\}$,
- $R^t_i = \{(s, s') \mid s' \in r^j_i(s, t)\ \}$,

- $V^t(s) = \upsilon^j(s, t)$.

In other words, W is the set of partial states $S(D_j(P))$, while the states that are accessible from a state s for an agent i are the possible states s' that are in $r_i^j(s, t)$. Finally, the valuation associated to a state s in this structure is the state $\upsilon^j(s, t)$ that deterministically follows from the possible initial state s and the action sequence up to $t - 1$.

Since each partial belief corresponds to a Kripke structure, a decomposed joint belief corresponds to a collection of Kripke structures. The evaluation of the truth value of objective and epistemic formulas ϕ, with $B_j(t)$ its corresponding partial belief, is then done as defined in Definition 4.3.2, where the Kripke model $\mathcal{K}$ is replaced by the Kripke model $\mathcal{K}_j(t)$ of the partial belief $B_j(t)$.

7.4. Agent's view

In the case of on-line multi-agent problems, we also need to keep track of what an agent *actually* knows. The set $S_i(t)$ in Section 5.4.3, representing the set of states that agent i considers possible based on his actual observations, is replaced with:

$$S_i^D(t) = \{ \ S_i^1, S_i^2, ..., S_i^n \ \}$$

where S_i^j is the set of partial states of the partial belief $B_j(t)$ that agent i considers possible at time t.

The reason is that we do not have a set of possible states anymore, but rather a set of sets of possible *partial* states. While in the on-line formulation of Chapter 5, a problem P had one hidden, true state, in this formulation the hidden, true state of the problem P is a vector of *partial* states: one partial state for each partial belief $B_j(t) \in B_D(t)$.

In other words, $S_i^D(t)$ is used to keep track of the partial states agent i considers possible in each partial belief that resulted from the decomposition. Through these sets we can evaluate the truth value of epistemic formulas such as $K_i\phi$.

Theorem 8. *$B_D(t) \models K_i\phi$ iff $\mathcal{K}_j(t), s_0 \models \phi, \forall s_0 \in S_i^j(t)$, where $\mathcal{K}_j(t)$ is the Kripke structure for the partial joint belief $B_j(t)$ that ϕ corresponds to.*

7.5. Example

We will use a version of the *Collaboration through Communication* problem from Chapter 5, with three blocks in total, in order to illustrate the decomposition approach.

Formally we have the planning problem $P = \langle A, F, I, O, N, S, G\rangle$, where $A = \{a, b\}$, $F = \{at(x, p_{k'}), in(b_i, p_k)\}$, $x \in A$, $k' \in [1, 4]$, $k \in \{1, 3, 4\}$, $i \in [1, 3]$, $I = \{at(a, p_2), at(b, p_2), \neg at(a, p_k), \neg at(b, p_k)\} \cup M$, where M contains the formulas expressing that each block has a unique location and $k = \{1, 3, 4\}$. The physical actions in O are $right_x$ and $left_x$, for each agent $x \in A$, and the set of sensors is empty. The set of sensing actions is $N = \{$**sense**$(x, [K_y in(b_i, p_k]), sense[x](in(b_i, p_k))\}$, for $x, y \in A$, $x \neq y$, $i \in [1, 3]$, $k \in \{1, 3, 4\}$, and the actions $sense[x](in(b_i, p_k))$ have precondition $at(x, p_k)$ and the owner is agent x. The goal is $G = (\vee_{k=1,4} K_a at(b_1, p_k)) \wedge (\vee_{k=1,4} K_b at(b_2, p_k))$. In total, there are four physical actions and 36 sensing actions.

7.5.1. Decomposition

First, we need to compute the initial decomposition Q, based on the initial formula I.

$$I = \{at(a, p_2), at(b, p_2), \neg at(a, p_1), \neg at(a, p_3), \neg at(a, p_4), \\ \neg at(b, p_1), \neg at(b, p_3), \neg at(b, p_4)\} \cup M \quad (7.1)$$

where M is:

$$\bigcup_{i=[1,4]} \{in(b_i, p_1) \vee in(b_i, p_3) \vee in(b_i, p_4), \neg in(b_i, p_1) \vee \neg in(b_i, p_3), \\ \neg in(b_i, p_1) \vee \neg in(b_i, p_4), \neg in(b_i, p_3) \vee \neg in(b_i, p_4)\} \quad (7.2)$$

We can see that $I = PI(I)$, that is the clauses of I are already the prime implicates of I - the unit clauses cannot be resolved (or subsume) any other clause in I, the clauses in M that denote the unique position of a block either cannot be resolved with other clauses or lead to tautologies.

Based on $PI(I)$, we have the initial decomposition $Q(P)$:

$$Q(P) = \{Q_1(P), ..., Q_{12}(P)\}$$

where

- $Q_1(P)$ to $Q_8(P)$ stand for each unit clause in $PI(I)$
- $Q_9(P) = \{in(b_1, p_1), in(b_1, p_3), in(b_1, p_4)\}$.
- $Q_{10}(P) = \{in(b_2, p_1), in(b_2, p_3), in(b_2, p_4)\}$.
- $Q_{11}(P) = \{in(b_3, p_1), in(b_3, p_3), in(b_3, p_4)\}$.

All the unit clauses in $PI(I)$ contain common literals: their truth value is known initially to all agents and all the conditional effects in which they appear are composed by common literals (in actions $right_x$ and $left_x$ where the conditional effects are of the form $at(a, p_x) \rightarrow at(a, p_{x-1}), \neg at(a, p_x)$ etc).

We update the initial decomposition $Q(P)$ to

$$Q(P) = \{Q_1(P), Q_2(P), Q_3(P)\}$$

where

- $Q_1(P) = \{in(b_1, p_1), in(b_1, p_3), in(b_1, p_4)\} \cup M$.
- $Q_2(P) = \{in(b_2, p_1), in(b_2, p_3), in(b_2, p_4)\} \cup M$.
- $Q_3(P) = \{in(b_3, p_1), in(b_3, p_3), in(b_3, p_4)\} \cup M$.

with $M = \{at(a, p_1), ..., at(a, p_4), at(b, p_1), ..., at(b, p_4)\}$.

The next step is to compute the dynamic decomposition $D(P)$ based on $Q(P)$, the sensing and physical actions.

First, we have the sensing actions **sense**$[x](in(b_i, p_k))$ with precondition $at(x, p_k)$. Since the sensed formula and the precondition contain only one literal, there is no dynamic relevance between any pair of sets $Q_j(P), Q_k(P) \in Q(P)$.

Secondly, we have the sensing actions **sense**$(x, [K_y in(b_i, p_k)])$, which contain only one objective literal $(in(b_i, p_k))$, which means that there is no dynamic relevance between any pair of sets $Q_j(P), Q_k(P) \in Q(P)$.

Lastly, we have the two physical actions $right_x$ and $left_x$. Each of these actions have conditional effects, denoting that agent x changes his position. Action $right_x$ has $e : at(x, p_k) \rightarrow \neg at(x, p_k), at(x, p_{k+1})$, for $k = \{1, 2, 3\}$, and action $left_x$ has $e : at(x, p_k) \rightarrow \neg at(x, p_k), at(x, p_{k-1})$, for $k = \{2, 3, 4\}$. The conditional effects of

$right_x$ and $left_x$ contain only common literals, so there is no dynamic relevance between any pair of sets $Q_j(P), Q_k(P) \in Q(P)$.

Based on the actions, we get the decomposition D:

$$D(P) = \{D_1(P), D_2(P), D_3(P)\}$$

where

- $D_1(P) = \{in(b_1, p_1), in(b_1, p_3), in(b_1, p_4)\} \cup M$.
- $D_2(P) = \{in(b_2, p_1), in(b_2, p_3), in(b_2, p_4)\} \cup M$.
- $D_3(P) = \{in(b_3, p_1), in(b_3, p_3), in(b_3, p_4)\} \cup M$.

with M as defined previously.

Each $D_j(P) \in D(P)$ defines a $I(D_j(P))$ which has 12 clauses (8 unit clauses denoting the position of the agents and 4 unit clauses denoting the possible positions of one block) and a $S(D_j(P))$ which contains 3 states (for the three possible positions of the block).

7.5.2. Belief Representation

Decomposition $D(P)$ allows us to define the decomposed, joint belief $B_D(t)$:

$$B_D(t) = \{\ B_1(t), B_2(t), B_3(t)\ \}$$

with:

- $B_1(t) = \{B_1(s_1, t), B_1(s_2, t), B_1(s_3, t)\}$
- $B_2(t) = \{B_2(s_1, t), B_2(s_2, t), B_2(s_3, t)\}$
- $B_3(t) = \{B_3(s_1, t), B_3(s_2, t), B_3(s_3, t)\}$

and, as an example:

$$B_2(s_1, t) = \langle v^2(s_1, t), r_a^2(s_1, t), r_b^2(s_1, t)\rangle$$
$$v^2(s_1, t) = \{in(b_2, p_1), \neg in(b_2, p_3), \neg in(b_2, p_4), at(a, p_2), at(b, p_2), \neg at(x, p_k)\}$$
$$r_a(s_2, t) = r_b(s_2, t) = \{s_1, s_2, s_3\}$$

for $t = 0, x \in A, k \in \{1, 3, 4\}$.

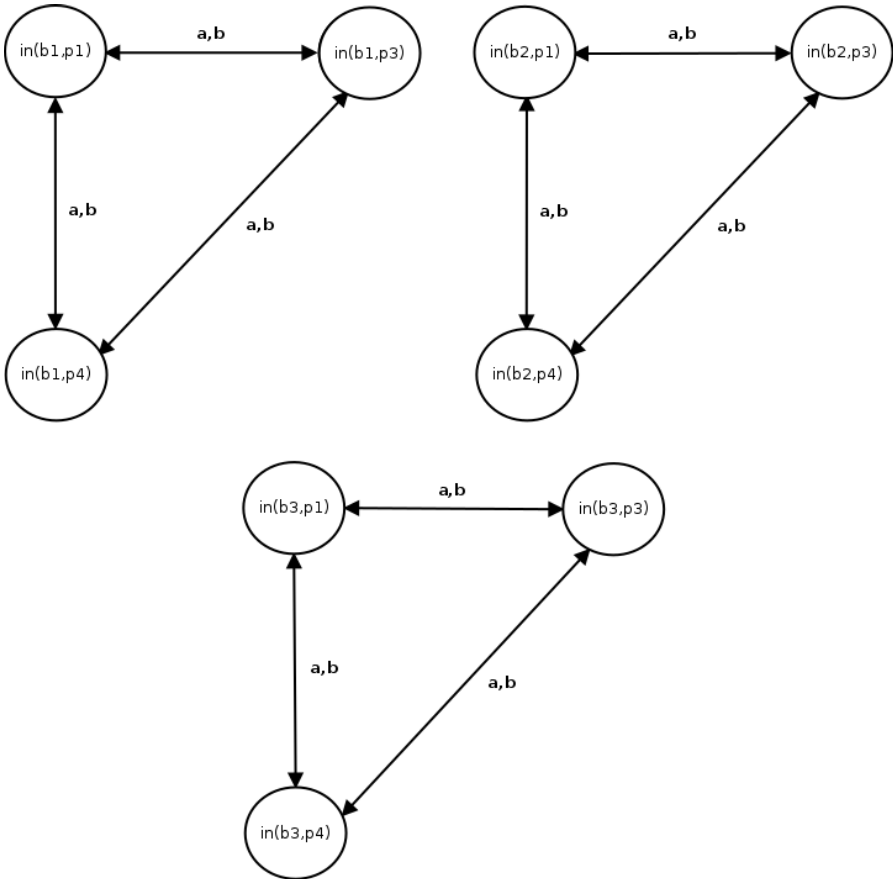

Figure 7.1: The Kripke stuctures that correspond to $B_D(0)$ for the *Collaboration-through-communication* example. From left to right, and top to bottom, the Kripke structures correspond to $B_1(0)$, $B_2(0)$ and $B_3(0)$. Reflexive relations and agents' positions are omitted.

In Figure 7.1, we can see the Kripke structures that correspond to $B_D(t)$, for $t = 0$. As we noted, the decomposed, joint belief $B_D(t)$ corresponds to a joint belief $B(t)$, as presented in Chapters 4 and 5. The Kripke structure of $B(t)$ for the same problem would resemble a complete graph with 27 nodes (states).

7.6. Translation into Classical Planning

The language for the translation $P' = K(P, B_D(t), S_i^D(t))$ is STRIPS extended with *negation*, *conditional effects*, and *axioms*.

The translation P' is based on the decomposed joint belief $B_D(t)$. For *every partial belief* $B_j(t) \in B_D(t)$ (with its corresponding decomposition $D_j(P)$ and the set of states $S(D_j(P))$ it defines) we have that P' contains:

- $L/(s, j)$ for encoding the states $v(s,t) \in B_j(s,t)$, for each $L \in D_j(P)$ and $s \in S(D_j(P))$, that express that the objective literal L is true in the current state s of the partial belief $B_j(t)$.
- $D_i(s, s')/j$ for encoding the sets $r_j(s,t) \in B_j(s,t)$, for all $s, s' \in S(D_j(P))$ and $j \in A$ that express that agent i can distinguish between the states s and s' of the partial belief $B_j(t)$.
- $T(s, j)$ for each $s \in D_j(t)$, that express that s is the *assumed true initial state* of the partial belief $B_j(t)$, and atoms $D_i(s, j)$ for representing that $s \notin S_i^j(t)$.

Formulas appearing in action preconditions, goals, and sensing expressions in P are assumed to be all literals or conjunctions of possibly epistemic literals L. A positive epistemic literal is an objective literal preceded by a sequence of epistemic operators possibly separated by negations, like $K_a \neg K_b K_c p$. The axioms in the translation are used to maintain the truth of epistemic literals.

We denote the set of objective literals in P as $L_F(P)$, the set of positive epistemic literals in P as $L_K(P)$, and the set of positive epistemic literals L that are suffixes of literals in $L_K(P)$ as $L_X(P)$. All objective and epistemic literals L will be characterized by the unique partial belief $B_j(t)$ to which they belong.

The literals ϕ/t in the translation are used to encode the truth of formulas ϕ in the assumed initial state of each partial belief; i.e., ϕ/t iff $\phi/(s, j)$ and $T(s, j)$, such that $\mathcal{L}(\phi) \subseteq D_j(P)$. Such formulas ϕ are the ones appearing in sensing and preconditions. The actions in $K(P, B_D(t),\ S_i^D(t))$ comprise the physical actions in P, the auxiliary actions $assume(s, j)$ for guessing the initial state in each partial belief $B_j(t)$, the action $\mathcal{E}$ for capturing the effects of passive sensing, and the sensing actions **sense**$[A](\phi)$ in P.

The actions $assume(s,j)$ must appear first in any plan for some possible s, excluding all other $assume(s',j)$ actions from being applied, for all $B_j(t) \in B_D(t)$. We assume that the goal G is a formula in conjunctive normal form, where objective and epistemic literals can appear.

Definition 7.6.1. The *classical problem* $K(P, B_D(t), S_\alpha^D(t)) = \langle F', I', O', G', X' \rangle$ with axioms, where α is the planning agent and $P = \langle A, F, I, O, N, S, G \rangle$ is such that:

- $F' = \{L/(s,j) : L \in L_F(P), L \in D_j(P), s \in S(D_j(P))\} \cup \{T(s,j) : s \in S(D_j(P)), D_j(P) \in D(P)\} \cup \{D_i(s,s')/j : i \in A, s, s' \in S(D_j(P)), D_j(P) \in D(P)\} \cup \{D_\alpha(s,j) : s \in S(D_j(P)), D_j(P) \in D(P)\}$,
- $I' = \{L/(s,j) : L \in L_F(P), L \in D_j(P), s \in S(D_j(P)), s \models L\} \cup \{D_\alpha(s,j) : s \in S(D_j(P)), s \notin S_\alpha^j(t)\} \cup \{D_i(s,s')/j : s, s' \in S(D_j(P)), D_j(P) \in D(P), s \notin r_i^j(s',t), r_i^j(s',t) \in B_j(t), i \in A\}$
- $G' = \bigwedge_{C \in G, \mathcal{L}(C) \subseteq D_j(P))} (D_\alpha(s,j) \vee C/(s,j)$, for all $s \in S(D_j(P))$
- Axioms X':
  - $K_i L/(s,j)$ iff $\wedge_{s' \in S(D_j(P))}[L/(s',j) \vee D_i(s,s')/j]$, for $K_i L \in L_X(P)$ and $\mathcal{L}(L) \subseteq D_j(P), D_j(P) \in D(P)$.
  - ϕ/t iff $\wedge_{s \in S(D_j(P))}[\neg T(s,j) \vee \phi/(s,j)]$, for $\mathcal{L}(\phi) \subseteq D_j(P)$
- Actions O':
  - **auxiliary actions** $assume(s,j)$, for all $s \in D_j(P)$ and $D_j(P) \in D(P)$, with prec. $\neg D_\alpha(s,j)$ and effect $T(s,j)$,
  - **physical actions** $a \in O$ owned by l have prec. $K_l Pre(a)/t$ and effects
    - $\neg K_l Pre(a)/(s',j) \rightarrow D_i(s,s')/j \wedge D_\alpha(s,j)$, for $Prec(a) \subseteq D_j(P)$, $D_j(P) \in D(P)$, $i \in A$ and $s, s' \in S(D_j(P))$,
    - for each $C \rightarrow E$ that does not contain only common literals, we have $C/(s,j) \rightarrow E/(s,j)$ for $s \in S(D_j(P))$, where $D_j(P) \in D(P), \mathcal{L}(C \cup E) \subseteq D_j(P)$. otherwise:
    - $C/(s,j) \rightarrow E/(s,j)$ for $s \in S(D_j(P))$, where $D_j(P) \in D(P)$,
  - all **sensing actions** e with owner l and preconditions $Pre(e)$ are mapped to actions with precondition $K_l Pre(e)/t$, and effect:
    - $\neg K_l Pre(e)/(s',j) \rightarrow D_i(s,s')/j \wedge D_\alpha(s,j)$, for $Prec(a) \subseteq D_j(P)$, $D_j(P) \in D(P)$, $i \in A$ and $s, s' \in S(D_j(P))$,

- **sensing actions** $e = \textbf{sense}[B](\phi) \in N$ with $\alpha \notin B$ are mapped into the same actions, with effects:
  - $\phi/(s,j) \wedge \neg\phi/(s',j) \rightarrow D_i(s,s')/j, D_i(s',s)/j$ where $\mathcal{L}(\phi) \subseteq D_j(P)$, and s, s' in $S(D_j(P))$, $i \in B$,
- **sensing actions sense**$[B](\phi) \in N$ with $\alpha \in B$ mapped into the same actions, with the effects
  - $\phi/(s,j) \wedge \neg\phi/(s',j) \rightarrow D_i(s,s')/j, D_i(s',s)/j$
  - $\phi/t \wedge \neg\phi/(s,j) \rightarrow D_\alpha(s,j)$,
  - $\neg\phi/t \wedge \phi/(s,j) \rightarrow D_\alpha(s,j)$, where $\mathcal{L}(\phi) \subseteq D_j(P)$, and for each pair s, s' in $S(D_j(P))$, $i \in B$,
- **auxiliary action** $\mathcal{E}$ that for each **psense**$[B](\phi)$ has effects
  - $\phi/(s,j) \wedge \neg\phi/(s',j) \rightarrow D_i(s,s')/j, D_i(s',s)/j$, for $i \in B$,
  - $\phi/t \wedge \neg\phi/(s',j) \rightarrow D_\alpha(s',j)$, if $\alpha \in B$,
  - $\neg\phi/t \wedge \phi/(s',j) \rightarrow D_\alpha(s',j)$, if $\alpha \in B$, where $\mathcal{L}(\phi) \subseteq D_j(P)$, and for each pair s, s' in $S(D_j(P))$.

The auxiliary actions $assume(s,j)$ have additional preconditions to enforce that only one partial state from each partial belief will be denoted as the assumed hidden partial state.

The translation is quadratic to the number of states of the largest partial belief and its properties are the same as in Section 5.5.1, where we replace the notion of assuming one state as the hidden true state ($assume(s)$) with the notion of assuming a partial state for each partial belief $B_j(t)$, denoted as $assume(s,j)$.

Definition 7.6.2 (Consistency)**.** Let π be a prefix of a plan for $P' = K(P, B_D(t), S_i^D(t))$. The normalized sequence $n(\pi)$ is *consistent* with the observations iff a) for any formula ϕ rendered *observable* by $n(\pi)$ at time t' from active or passive sensing, $B_j(t'), s \models \phi$ iff ϕ is observed to be true at time t', and $assume(s,j)$ is one of the $assume$ actions in π, where $B_j(t')$ is the partial belief that corresponds to ϕ, and b) the physical actions a in $n(\pi)$ are all applicable in P (i.e., owners know the preconditions).

Theorem 9 (Soundness)**.** *a) If π is plan for $K(P, B_D(t), S_i^D(t))$ that is consistent with the observations, the execution of $n(\pi)$ leads to the goal in the problem P. b) Otherwise, if π' is the shortest prefix of π that is inconsistent and π includes the action $assume(s,j)$, after the execution of $n(\pi')$ in P, $s \notin S_i^j(t'), S_i^j(t') \in S_i^D(t')$ where t' is* *the resulting time step and j indicates the partial belief $B_j(t)$ where the inconsistency occurred.*

Theorem 10 (Completeness)**.** *If s^* is the true hidden state in P, and there is an action sequence that achieves K_iG for an agent i, then there is a plan π for $K(P, B_D(t), S_i^D(t))$ that starts with the actions $assume(s, j)$, where s^* is the union of all s that appear in an $assume$ action in π, and any such plan is consistent.*

CHAPTER **8**

Examples and Experiments

8.1. Gossiping Problem

The Gossiping Problem was introduced in 1979 by (Entringer and Slater, 1979), It can defined as follows:

There are n agents each of which knows some secret not known to anybody else. Two agents can make a telephone call and exchange all secrets they know. How many calls does it take to share all secrets, i.e., how many calls have to take place until everybody knows all secrets?

If we represent the problem as a graph, where the nodes are agents and the edges indicate which pairs of agents can communicate with each other, we would get a complete graph. It is proven that the minimum number of calls needed for all agents to know all secrets is $2(n-2)$ (Harary and Schwenk, 1974).

Different variations of the problem exist depending on whether agents can communicate with all the other agents or with only specific ones, whether agents when communicating share all the secrets they know or only one, whether the communication is one way or both-ways etc (Hedetniemi et al., 1988). Variations also exist in terms of knowledge. The problem, as we defined it, necessitates that in the end all secrets are shared knowledge (all agents know all secrets) and there are variations where the agents need to achieve second-order shared knowledge (all agents know that all agents know all secrets) and so on (Herzig and Maffre, 2017).

We introduce a variation of the original Gossiping problem called Public Gossiping. Our

Public Gossiping	Decomposition	
Num. Agents	**h(add)**	**h(max)**
4	0.02	0.02
5	0.02	0.6
6	0.02	138
10	0.3	-

Table 8.1: Experimental results for the Public Gossiping problem with the decomposition approach, using A^* with the additive and the max heuristic. Times are in seconds.

variation introduces three additional properties to the original problem. First, all phone calls are public. This means that all agents are aware that a communication between two agents has occurred. Second, all sensors are public, which, in combination with all phonecalls being public, means that at every time t all agents know which secrets the other agents know. Third, when a pair of agents communicate they share their knowledge about all secrets.

The Public Gossiping planning problem then is $P = \langle A, F, I, O, N, S, G \rangle$ (Kominis and Geffner, 2015), where A is the set of agent names $(1, ..., n+1)$, $F = \{p_1, ..., p_n\}$ the secrets, I represents the initial situation which is $\bigwedge_{i \in \{1,...,n\}}(p_i \vee \neg p_i)$, O is the set of *physical actions* which is empty, N is the set of *sensing actions*, where for each pair of agents $i, j \in A \setminus \{(n+1)\}$, we have a parallel sensing action $sense[i, j](K_i p_1, ..., K_i p_n, K_j p_1, ..., K_j p_n)$ indicating that the agents share their knowledge about all secrets, S is the set of (passive) *sensors*, where for each agent $i \in A \setminus \{(n+1)\}$ we have a passive sensor $psense[i](p_i)$ indicating that each agent knows initially one secret, and G is the goal where $G = \bigwedge_{i,j \in \{1...n\}}(K_i p_j \vee K_i \neg p_j)$, indicating that all agents know all secrets. Agent $(n+1)$ takes the role of the planning agent. This allows us to solve the problem in just one planning phase: since there are no physical actions and the planning agent does not have a passive sensor, nor does he participate in any sensing action, if a plan is found then it cannot fail during execution and the goal will be achieved.

In Table 8.1 we can see the experimental results for the Public Gossiping problem with the decomposition approach, using the additive and max heuristics. The size of the decomposition $D(P)$ is equal to n, one decomposition per each secret, and all decompositions $D_j(P) \in D(P)$ have the same number of initial states which is 2 (the two different truth values each secret can take). The length of the plans, in the case where the maximum heuristic was used, was, as predicted, $2(n-2)$, where n the number of agents without considering the planning agent. In the cases where the additive heuristic was used, the length was $2(n-2)+1$.

Problems	Linear translation	Decomposition
Active Muddy Child	✓	✗
Sum	✓	✗
Wordrooms	✓	✗
Collaboration-through-Comm	✓	✓
Meeting Problem	✗	✗
Situated Dialogue	✗	✓
Lights	✗	✗
Public Gossiping	✗	✓

Table 8.2: Planning problems for which the linear translations and the decomposition approach can be used.

8.2. Experiments

The linear translation and the decomposition approach cannot be used for all the problems we have seen in Chapters 4 and 5. The linear translation depends on an additional requirement that not all problems satisfy, while the decomposition approach relies on the structure of each problem. Table 8.2 shows the problems on which the two methods can be applied.

Concerning the linear translation, we can see that the Meeting Problem, the Situated Dialogue problem, the Lights and the Public Gossiping problem do not fulfill the requirements. In the first two problems, agents sense non-static objective literals. In the first case, whether an agent is in the same position with a landmark, in the second case whether a block that can be moved has a spatial relationship with a hidden block. The Public Gossiping problem has sensing of epistemic literals that do not involve all agents (only pair of agents) and the Lights problem has both: sensing of non-static objective literals (lights that can be turned on and off) and sensing of epistemic literals that do not involve all agents.

Concerning the decomposition approach, the Active Muddy Child does not have a decomposition due to the fact that the initial uncertainty is given by a formula of one clause that contains all literals (representing the fact that at least one of them is muddy). Even if we added this information as a sensing action, simulating the fact that all agents get to know it during execution (assuming the hidden true state is not the one where all agents are clean), the problem would not have a decomposition since all literals would be dynamically relevant to each other due to the formula in that sensing action. Similarly, there is no decomposition for the Sum problem, the Wordrooms, the Meeting problem

Problems	#States	Decomposition
Collaboration-through-Comm (3)	27	3 - 3 - 3
Collaboration-through-Comm (4)	81	3 - 3 - 3 - 3
Collaboration-through-Comm (5)	243	3 - 3 - 3 - 3 - 3
Situated Dialogue (2,5)	25	5 - 5
Situated Dialogue (4,5)	625	5 - 5 - 5 - 5
Situated Dialogue (2,6)	36	6 - 6
Public Gossiping (4)	16	2 - 2 - 2 - 2
Public Gossiping (6)	64	2 - 2 - 2 - 2 - 2 - 2

Table 8.3: For the Collaboration through Communication, the number in the parenthesis is the number of blocks in the problem. For the Situated Dialogue, (X,Y) stands for X blocks whose position is unknown, with Y possible positions each. For the Public Gossiping, the number in the parenthesis is the number of secrets. In column #States we see the number of possible initial states of the original problem. In the decomposition column $l_1 - l_2 - ... - l_n$ stands for n partial beliefs, where l_j is the number of possible initial partial states of the partial belief $B_j(t)$

and the Lights problem.

The three problems that do have a decomposition (Collaboration through Communication, Situated Dialogue and Public Gossiping) share a common property: the initial uncertainty the agents have about a certain object in the problem is independent of their uncertainty about the rest of the objects. Specifically, for Collaboration through Communication and Situated Dialogue, the possible positions of the blocks whose initial position is not known do not (and cannot) overlap. In the case of Public Gossiping, the truth values of the secrets are independent of each other.

In Table 8.3 we see the results of the decomposition approach in terms of the size the decomposed, joint belief $B_D(t)$. In the case of "Collaboration through communication", the size of the decomposed joint belief depends on the number of blocks, while the number of states of each partial belief on the possible positions of each block. The same is true for the Situated Dialogue. In the case of the Public Gossiping problem, each secret corresponds to a partial belief with two possible states (the two possible truth values of the proposition modeling the secret).

Table 8.4 shows the experimental results of our optimizations in comparison with the quadratic translation of Chapter 5. The results were obtained by using the FD planner as the classical planner (Helmert, 2006), the A^* algorithm with the additive heuristic, on a Linux machine at 2.93 GHz and 4GB of RAM.

Problems	# States	Quadratic	Linear	Decomposition
Active Muddy Child (5)	31	2.2	0.04	X
Active Muddy Child (6)	63	73	0.17	X
Active Muddy Child (7)	127	-	0.7	X
Coll-through-comm (3)	27	2.3	0.3	0.02
Coll-through-comm (4)	81	96	2.2	0.02
Coll-through-comm (5)	243	-	-	0.07
Situated Dialogue (2,5)	25	2.8	X	0.1
Situated Dialogue (2,6)	36	5.8	X	0.15
Public Gossiping (4)	16	0.92	X	0.02
Public Gossiping (5)	32	22	X	0.02
Public Gossiping (6)	64	660	X	0.02
Public Gossiping (10)	1024	-	X	0.26

Table 8.4: Experimental results of optimizations in comparison with the quadratic translation. Times are in seconds and an X denotes that the approach was not applicable in the problem.

PART IV

Conclusions

8.3. Contributions

In this section we outline the main contributions of this thesis.

- We introduced a framework for linear, multi-agent planning problems which allows the handling of arbitrary epistemic formulas. We show that this framework captures a fragment of dynamic epistemic logic ($\mathcal{S}5$) and, though simple, it provides a convenient modeling language, simple semantics and a computational model for reasoning with nested beliefs. Furthermore, we presented a translation to classical planning, which is based on similar translation approaches for single agent contingent and conformant planning. Such translation approaches take advantage of the work done in classical planning and allow for usage of off-the-shelf classical planners. The translation is sound and complete, and its complexity is quadratic in the number of initial states.

- We introduced a framework for on-line, multi-agent planning in partially observable environments. This framework is an extension of our work for linear, multi-agent planning problems, where we included the notion of a planning agent and we made explicit and formal the conditions under which an epistemic formula is true in an on-line setting. Furthermore, we provided an algorithm and a new translation to classical planning that allows us to compute a plan within a plan-execute-observe-replan cycle. The algorithm, assuming no dead ends, guarantees termination after a bounded number of calls to the classical planner. The translation itself introduces the paradigm where the planner chooses the next assumed state, in contrast with being provided with one externally.

- We identified two special cases that allow for more scalable approaches. The first case concerns problems where (i) the objective literals contained in sensing action/sensors are static, and (ii) the sensing actions that contain epistemic literals include all the agents of the problem. A translation to classical planning was presented for these problems and its complexity is linear to the number of initial states. The second case concerns problems which can be decomposed to sub-problems. We introduce the notion of relevance among propositional atoms which allows us to define a partial belief for each sub-problem, and provided a translation to classical planning which is quadratic to the number of initial states of the largest subproblem.

8.4. Ongoing and Future Work

8.4.1. A Syntactic Approach

Work has been done on a different type of approaches for multi-agent planning, most notably in (Cooper et al., 2016) and (Muise et al., 2015). These approaches do not make use of Kripke structures but they depend on formulas that contain epistemic literals. This means that the knowledge (or beliefs) the agents possess is explicitly represented in the formula with the usage of propositions K_iL (or B_iL).

We have preliminary work concerning a similar syntactic approach. The aim of our approach is to have a framework which is independent of Kripke structures and allows us to evaluate (objective and epistemic) queries on an objective formula. This is the key difference between our formulation and similar syntatic approaches: the knowledge of the agents is not expressed within the formula itself but it is compiled into propositional logic.

To achieve this, we keep track the set of possible states at time t, which is represented by an objective formula $\phi(t)$ and for each agent we keep track of the set of literals $S_i(t)$ that he has sensed up to time t. The assumptions we make about the problems we deal with are that there are no physical actions, plus the additional requirements that are needed for the linear translation.

More specifically, given $\phi(t)$ and $S_i(t)$, we can efficiently:

1. evaluate the truth value of any epistemic literal K_iL where L is an objective literal, and
2. compute an objective formula ψ that identifies all and only the states where K_iL would be evaluated to true.

Both operations, the evaluation of epistemic literals and the computation of ψ, are done in time polynomial to the size of $PI(\phi(t))$. The fact that we can compute the objective formula ψ allows us to define how communication affects the knowledge of agents. Intuitevely, since ψ identifies all and only the states where K_iL is true, announcing K_iL at time t means excluding from the set of all possible states at time $t-1$ the states where K_iL is not true. This is the same as $\phi(t) = \phi(t-1) \wedge \psi$, since ψ identifies only the states K_iL is true.

The following example is indicative of how we evaluate epistemic literals.

Example 8.4.1. *Suppose* $PI(\phi(0)) = \{(a \vee b), (a \vee c)\}$, $S_i(t) = \{b, c\}$ *and the hidden* *true state $s^* = \{a, \neg b, c\}$. The operation of evaluating $K_1 a$ is achieved by adding to $PI(\phi(0))$ all the truth values of the literals that agent 1 has sensed (which excludes all states that are not possible base on the knowledge agent 1 has about the true state) and apply unit resolution. The unit clause $(\neg b)$ will be resolved with the clause $(a \vee b)$, resulting to (a), which means that $K_1 a$.*

In other words, since we have $K_1(a \vee b)$ (because in all states $a \vee b$ is true, agent 1 knows it), which can be written as $K_1(\neg b \rightarrow a)$, and we know that $K_1 \neg b$ (because he has sensed it), we can derive that $K_1 a$ is true without actually representing the knowledge of the agent with epistemic literals.

The operation of computing the formula ψ that corresponds to $K_i L$, where L is objective, is the following:

1. if agent i has sensed L, then $\psi = L$, otherwise
2. identify all clauses $C_1 ... C_n$ that contain L, $C_k = L \vee C'_k$, since at least one of these must be resolved to obtain L,
3. from all subclauses $C'_1 ... C'_n$ identify the ones which contain only propositions which have been sensed by i ($C'_1 ... C'_m$) since only those propositions can be used for resolution and, possibly, provide L,
4. agent 1 must know the negation of at least one of these subclauses ($\neg C'_1 \vee ... \vee \neg C'_m$), since at least one clause needs to be resolved to the point that only L remains,

In our example, the objective formula that corresponds to $K_1 a$ is $\psi = \neg b \vee \neg c$, and announcing $K_1 a$ means adding ψ to the original formula, which results in $\phi(1) = (a \vee b) \wedge (a \vee c) \wedge (\neg b \vee \neg c)$, while announcing $\neg K_1 a$ means adding $\neg \psi$ to the original formula, which results in $\phi(1) = (a \vee b) \wedge (a \vee c) \wedge (b) \wedge (c)$

Concerning the nested epistemic literals, e.g. $K_i K_j L$, the approach is similar. For i to know $K_j L$ it means that i can distinguish between some states where $K_j L$ is true and all the states where it is false. But, all the states where $K_j L$ is true are identified by ψ, which means that $K_i K_j L$ is evaluated as true only in the states where $K_i \psi$ is evaluated as true. In turn, $K_i \psi$ corresponds to a new objective formula ψ' which can be used as defined above [1].

The operations are done in polynomial time given that we have the prime implicates

[1] Additional definitions are necessary concerning the computation of the formula that corresponds to $K_i \Phi$, where Φ is a formula in CNF. First, that K_i is distributable over conjunctions, and second, how we deal with $K_i(l_1 \vee ... \vee l_n)$. Mainly, we need to compute the formula that corresponds to every subset of $l_1 \vee ... \vee l_n$ without taking into consideration the literals l_k that are sensed by i, which are treated individually.

of formula $\phi(t)$. Unfortunately, the number of the prime implicates of a formula ϕ can be exponential to the size of ϕ, and we need to compute them *every time* there is a communication between the agents. Nevertheless, computing the prime implicates of $PI(\Phi) \wedge \Psi$ can be done in polynomial time as long as Ψ consists of unit clauses and/or clauses that contain only *pure literals* (literals that appear only negated or only positive in the entire formula). This allows us to identify special cases where the formulation can be used efficiently:

- when the announcement consists only of sensed literals, thus adding only unit clauses
- when the announcement consists of clauses that cannot be resolved with any other clause, or resolutions lead to tautologies

The "Situated Dialogue" problem falls under the first case: agents communicate literals they have sensed. The problems "Collaboration through Communication" and "Word-rooms" have plans that fall in the same category. The "Muddy Children" and "Active Muddy Child" problems fall under the second case: all the announcements before the one that achieves the goal have formulas that contain only pure literals.

We want to study further such syntactic formulations. Our objective is to have an approach that will allow us to solve meaningful problems without having the restriction of recomputing the prime implicates of a formula multiple times.We expect such formulations to be sound but not complete.

8.4.2. Incorporating Belief

The belief representation $B(t)$ and its extension for on-line, planning problems deals with knowledge. Although knowledge is sufficient for many meaningful cooperative planning problems, we would like to be able to incorporate belief: agents may believe something to be true while it is actually false. As an example, imagine an agent who leaves in his drawer a book and exits the room. While he is away, another agent enters the room, takes the book from the drawer and hides it in the closet. When the first agent returns to the room, he believes that the book is still in the drawer, since he does not know about the actions of the second agent. At the same time, the second agent knows that the book is not in the drawer but in the closet.

Extending our formulation to incorporate belief would allow the relaxation of the assumptions we make concerning the problems we deal with. Specifically, the requirement that all actions (physical and sensing) and sensors are public. Dealing with beliefs will further allow us to tackle problems where communication is noisy or restricted due to resources, as well as problems where agents have information that do not want to share and/or private goals.

Besides the question of how to incorporate beliefs in our formulation and at the same time be scalable, a number of questions arise as to how we should treat knowledge and beliefs. Planning problems are about achieving a goal with certainty. An agent believing that a goal is achieved is not enough, the agent must *know* that the goal is achieved. This implies that we need a framework which allows knowledge and belief to co-exist (Kraus and Lehmann, 1988; van der Hoek, 1990), which in terms of automated planning would also need to address an additional question: when does a belief become knowledge and when knowledge becomes a belief in terms of actions being applied.

APPENDIX **A**

Properties of $K(P)$ translation

In the following proofs, ϕ/s stands for a formula ϕ where every objective and epistemic atom is tagged with the state s. For example, if $\phi = (p_1 \vee ... \vee p_n) \wedge (K_i p \vee K_i \neg p)$, then $\phi/s = (p_1/s \vee ... \vee p_n/s) \wedge (K_i p/s \vee K_i \neg p/s)$.

Lemma A.1. *Suppose an action sequence π which is applicable in both P and $K(P)$. If π achieves L/s, where L is an objective literal, in $K(P)$, then π achieves L in the state s of P, $s \in b_I$.*

Proof. Suppose π is empty. This means $L/s \in I'$ which is true only if $I \models s$ and $s \models L$, $s \in b_I$.

Suppose $\pi = \pi', \alpha$. Two cases to consider: (i) action α after π' achieves L/s, or (ii) π' achieves L/s and α does not delete it.

If (i) is true, then α in P must contain a conditional effect $C \rightarrow L$, and by inductive hypothesis, π must achieve C in state s of P. Therefore, α achieves L in state s of P. If (ii) is true, by inductive hypothesis, it means that π' achieves L, and for every effect $C' \rightarrow \neg L$, π' achieves $\neg L'$ in state s of P, for some literal L' in C', and thus, π must achieve L in state s of P too.

□

Translation $K(P)$ introduces atoms $D_i(s, s')$ that represent the fact that $s' \notin r_i(s, t)$. These atoms, after they have been achieved, cannot become false. This means that all the uncertainty the agents have is due to the initial set of states and this uncertainty is monotonically decreasing. This is evident in our formulation $B(t)$ from the fact that there is no action whose effect is to *add* a state s to a set $r_i(s', t)$ - there are only actions that remove states that disagree with s' in the truth value of some formula ϕ. Similarly, in the $K(P)$ translation, atoms $D_i(s, s')$ do not appear negated in the effect of an action.

Lemma A.2. *Suppose an action sequence π which is applicable in both P and K(P). If π achieves $D_i(s, s')$ in K(P), then π achieves $s' \notin r_i(s, t)$ in P, $s, s' \in b_I$.*

Proof. If π is empty it cannot achieve $D_i(s, s')$ since for $t = 0$ we have $r_i(s, t) = b_I$ for $s \in b_I$.

Suppose $\pi = \pi', \alpha$. There are two cases to consider: (i) action α is a sensing action (or an update action) that achieves $D_i(s, s')$, or (ii) π' achieves $D_i(s, s')$ and action α does not delete it.

If (i) is true, then α has a conditional effect with condition $\phi/s \wedge \neg\phi/s'$ (or $\neg\phi/s'$) and effect $D_i(s, s')$. This means that π' achieves $\phi/s \wedge \neg\phi/s'$ (or $\neg\phi/s'$), and by Lemma A.1, π' achieves $s \models \phi$ and $s' \not\models \phi$ (or just $s' \not\models \phi$) in P. Therefore π achieves $s' \notin r_i(s, t)$ in P. If (ii) the proof is direct since there is no action that can delete $D_i(s, s')$. □

Lemma A.3. *Suppose an action sequence π which is applicable in both P and K(P). If π achieves $K_i L/s$ in K(P), and L is an objective literal, then π achieves $B(t), s \models K_i L$ in P.*

Proof. An epistemic literal $K_i L/s$ is true when the axiom $\langle K_i L/s, L/s \wedge \bigwedge_{s' \in b_I} (L/s' \vee D_i(s, s'))\rangle$ is true.

Since π achieved $K_i L/s$ we have that L/s is achieved, and for all states $s' \in b_I$, either L/s' is achieved or $D_i(s, s')$ is achieved.

Suppose that $B(t), s \not\models K_i L$. It must be that there exists an $s' \in b_I$ such that (i) $B(t), s' \not\models L$ and (ii) $s' \in r_i(s, t)$. By Lemma A.1 we have that if π achieves L/s', then $s' \models L$ in P. By Lemma A.2 we have that if π achieves $D_i(s, s')$, then $s' \notin r_i(s, t)$ in P. Therefore, π must achieve $B(t), s \models K_i L$. □

Lemma A.4. *Suppose an action sequence π which is applicable in both P and K(P). If π achieves L in K(P), then π achieves $B(t) \models L$ in P.*

Proof. An atom L is true when the axiom $\langle L, \wedge_{s \in b_I} (L/s \vee D_i(s, s))\rangle$ is true.

Since π achieved L we have that for all states $s \in b_I$, either L/s is achieved or $D_i(s, s)$ is achieved.

Suppose that $B(t) \not\models L$. It must be that there exists an $s \in b_I$ such that (i) $B(t), s \not\models L$ and (ii) $s \in r_i(s, t)$. By Lemma A.1 we have that if π achieves L/s, then $s \models L$ in P. By Lemma A.2 we have that if π achieves $D_i(s, s)$, then $s \notin r_i(s, t)$ in P. Therefore, π must achieve $B(t), s \models L$ for all possible states in P, and as a result $B(t) \models L$. □

Lemma A.5. *If an action sequence π is applicable in $K(P)$, then π is applicable in P.*

Proof. If π is empty it is trivial. Suppose $\pi = \pi', \alpha$. Since π is applicable in $K(P)$, π' is applicable in $K(P)$, and by inductive hypothesis π' is applicable in P. Furthermore, since π is applicable in $K(P)$, then π' achieves L for $L \in Pre(\alpha)$. By Lemma A.4, π' achieves L in all states s that are possible in P. Therefore, π is applicable in P. □

Theorem A.1. *If an action sequence π is a plan for $K(P)$, then π is a plan for P.*

Proof. Direct from previous lemma. Consider a problem P' which is the same as P, with the addition of an action α' which has precondition the goal G of P. If π is a plan for $K(P)$, then $\pi' = \pi, \alpha'$ is a plan for $K(P')$, and by the previous lemma, π' is applicable in P, thus, it achieves G and π is a plan for P. □

Lemma A.6. *Suppose an action sequence π which is applicable in both P and $K(P)$. If π achieves L, where L is an objective literal, in some possible initial state s of P, $s \in b_I$, then π achieves L/s in $K(P)$.*

Proof. Suppose π is empty. Since L is true in s and $I \models s$, L/s must be true in $K(P)$.

Suppose $\pi = \pi', \alpha$. Since π achieves L in state s of P, then (i) α is a physical action with a conditional effect $C \to L$, and π' must have achieved C in s, or (ii) π achieves L in state s of P, and for any conditional effect of α, $C' \to \neg L$, π achieves $\neg L'$ in state s of P, for some $L' \in C'$.

If (i) is true, then α in $K(P)$ must contain a conditional effect $C/s \to L/s$, and by inductive hypothesis, π must achieve C/s in $K(P)$, and, therefore L/s. If (ii) is true, by inductive hypothesis, π achieves L/s and $\neg L'/s$, for some $L'/s \in C'/s$, so π achieves $\neg C'/s$ and L/s in $K(P)$. □

Lemma A.7. *Suppose an action sequence π which is applicable in both P and K(P). If π achieves $s' \notin r_i(s, t)$ in P, $s \in b_I$, then π achieves $D_i(s, s')$ in $K(P)$.*

Proof. If π is empty it cannot achieve $s \notin r_i(s,t)$ since for $t = 0$ we have $r_i(s,t) = b_I$ for $s \in b_I$.

Suppose $\pi = \pi', \alpha$. Since α is a sensing action (or an update action) that achieves $s' \notin r_i(s,t)$, π' achieves $s \models \phi$ and $s' \not\models \phi$ (or just $s' \not\models \phi$), and, by Lemma A.6, π' achieves ϕ/s and $\neg\phi/s'$ (or just $\neg\phi/s'$) in $K(P)$. Action α has a conditional effect $\phi/s \wedge \neg\phi/s' \rightarrow D_i(s,s')$ (or $\neg\phi/s' \rightarrow D_i(s,s')$), therefore π achieves $D_i(s,s')$. □

Lemma A.8. *Suppose an action sequence π which is applicable in both P and K(P). If π achieves $B(t), s \models K_i L$ in P, and L is an objective literal, then π achieves $K_i L/s$ in $K(P)$.*

Proof. Since π achieves $B(t), s \models K_i L$, then for all states $s' \in b_I$ either $s' \models L$ or $s' \notin r_i(s,t)$.

Suppose that π when applied in $K(P)$ does not achieve $K_i L/s$. This means that there exists a state s' such that $\neg L/s'$ and $\neg D_i(s,s')$. By Lemma A.6, if π achieves $s' \models L$ in P, then π achieves L/s' in $K(P)$, and by Lemma A.7, if π achieves $s' \notin r_i(s,t)$ in P, then π achieves $D_i(s,s')$ in $K(P)$. Therefore, π achieves $K_i L/s$ in $K(P)$. □

Lemma A.9. *Suppose an action sequence π which is applicable in both P and K(P). If π achieves $B(t) \models L$ in P, then π achieves L in $K(P)$.*

Proof. Atoms L are achieved in $K(P)$ through axioms of the form $\langle L, \bigwedge_{s \in b_I}(L/s \vee D_i(s,s))\rangle$.

Since π achieved $B(t) \models L$ we have that for all states $s \in b_I$, either $s \models L$ is achieved or $s \notin r_i(s,t)$ is achieved.

Suppose that, after the execution of π, L is false in $K(P)$. It must be that there exists an $s \in b_I$ such that (i) L/s has not been achieved, and (ii) $D_i(s,s)$ has not been achieved. By Lemma A.6 we have that if π achieves $s \models L$ in P, then L/s is true in $K(P)$. By Lemma A.7 we have that if π achieves $s \notin r_i(s,t)$ in P, then $D_i(s,s)$ is true in $K(P)$. Therefore, π must achieve L/s for all possible states in $K(P)$ at time $t = |\pi|$, and as a result, L is true. □

Lemma A.10. *If an action sequence π is applicable in P, then π is applicable in $K(P)$.*

Proof. If π is empty it is trivial. Suppose $\pi = \pi', \alpha$. Since π is applicable in P, π' achieves $s \models L$ for $L \in Pre(\alpha)$ and $s \in b_I$. By Lemma A.9, π must achieve L/s for $L \in Pre(\alpha)$ and $s \in b_I$ in $K(P)$. Therefore, π is applicable in $K(P)$. □

Theorem A.2. *If an action sequence π is a plan for P, then π is a plan for $K(P)$.*

Proof. Direct from previous lemmas. Consider a problem P' which is the same as P, with the addition of an action α' which has precondition the goal G of P. If π is a plan for P', then $\pi' = \pi, \alpha'$ is a plan for $K(P')$, and by the previous lemma, π' is applicable in $K(P)$, thus, it achieves G and π is a plan for $K(P)$. $\square$

Theorem 1. (p.51) An action sequence π is a plan that solves the linear multi-agent planning problem P iff π is a plan that solves the classical planning problem with axioms $K(P)$.

Proof. Direct from theorems A.1 and A.2. $\square$

APPENDIX **B**

Properties of $K(P, B(t), S_i(t))$ translation

From now on we will refer to the translation $K(P, B(t), S_i(t))$ as P'.

Definition B.1. Given a sequence of actions $\pi = assume(s), a_1, \epsilon, a_2, \epsilon..., a_n, \epsilon$, where ϵ is the auxiliary action representing the application of all passive sensors, the normalization of π is $n(\pi) = a_1, a_2, ..., a_n$.

Lemma B.1. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves L/s in P', then $n(\pi)$ achieves $s \models L$ in P.*

Proof. Suppose π, and by extension $n(\pi)$, is empty. Since L/s is true in I', then $I \models s$ and $s \models L$.

Suppose $\pi = \pi', \alpha$. There are two cases to consider: (i) action α after π' achieves L/s, or (ii) π' achieves L/s and α does not delete it.

If (i) is true, then α in P contains a conditional effect $C \rightarrow L$, and by inductive hypothesis, $n(\pi')$ must achieve C in state s of P. Therefore, $n(\pi)$ achieves L in state s of P. If (ii) is true, by inductive hypothesis, $n(\pi')$ achieves L and, for every conditional effect $C' \rightarrow \neg L$ in α, $n(\pi')$ achieves $\neg L'$ in state s of P, for a literal L' in C'. Therefore $n(\pi)$ achieves L is state s of P. $\square$

The $K(P, B(t), S_i(t))$ translation introduces atoms $D_i(s, s')$ and $D_i(s)$ that represent the fact that $s' \notin r_i(s,t)$ and $s \notin S_i(t)$. These atoms, after they have been achieved, cannot become false. For $D_i(s,s')$, this means that all the uncertainty the agents have is due to the initial set of states and this uncertainty is monotonically decreasing. For $D_i(s)$, this means that as long as a state is not consider possible at some time t due to an observation that disagrees with the assumed state, the state cannot become possible at a later time $t' > t$. This is evident in our formulation from the fact that there is no action whose effect is to *add* a state s to a set $r_i(s',t)$ or $S_i(t)$ - there are only actions that remove states that disagree with s' in the truth value of some formula ϕ, or states from $S_i(t)$. Similarly, in the translation, none of these atoms appear negated in the effect of an action.

Lemma B.2. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves $D_i(s,s')$ in P' due to a sensing action* **sense**$[i](\phi)$ *or a passive sensor* **psense**$[i](\phi)$*, where ϕ is an objective formula, then $n(\pi)$ achieves $s' \notin r_i(s,t)$ in P.*

Proof. Suppose π is empty. Since $D_i(s,s') \in I'$ then $s' \notin r_i(s,t)$.

Suppose $\pi = \pi', \alpha$. There are two cases to consider: (i) α is a sensing action **sense**$[i](\phi)$ (or a passive sensor **psense**$[i](\phi)$) that achieves $D_i(s,s')$, and (ii) π' achieves $D_i(s,s')$ and α does not delete it.

If (i) is true, α has a conditional effect $\phi/s \wedge \neg\phi/s' \to D_i(s,s')$, and π' has achieved ϕ/s and $\neg\phi/s'$. By Lemma B.1, $n(\pi')$ achieves $s \models \phi$ and $s' \not\models \phi$. Therefore, by definition of the sensing action update, we have $s' \notin r_i(s,t)$. If (ii) is true, it is direct due to the fact that no action can delete $D_i(s,s')$. □

Lemma B.3. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves K_iL/s in P', then $n(\pi)$ achieves $B(t), s \models K_iL$ in P.*

Proof: Epistemic literals K_iL/s are achieved through axioms of the form $\langle K_iL/s, \bigwedge_{s' \in b_I}(L/s' \vee D_i(s,s'))\rangle$.

Since π achieved K_iL/s, for all states $s' \in b_I$, either L/s' is achieved or $D_i(s,s')$ is achieved.

Suppose that $B(t), s \not\models K_iL$. It must be that there exists a state $s' \in b_I$ such that (i) $s' \not\models L$ and (ii) $s' \in r_i(s,t)$. By Lemma B.1 we have that if π achieves L/s', then $n(\pi)$ achieves $s \models L$ in P. By Lemma B.2 we have that if π achieves $D_i(s,s')$, then $n(\pi)$ achieves $s \notin r_i(s,t)$ in P. Since for all states, either $s \models L$ or $s \notin r_i(s,t)$ is achieved, there cannot be a state s' s.t. $s' \not\models L$ and $s' \in r_i(s,t)$. Therefore, $n(\pi)$ must achieve $B(t), s \models K_iL$.

Lemma B.4. *Suppose a sequence of actions π that is applicable in P' and the first action in π is $assume(s)$. We have that ϕ/t is achieved if and only if ϕ/s is achieved.*

Proof: Since there is only one $assume(s)$ action that appears in π, $s \in S_i(t)$, as the first action, and its effect is $T(s)$, then for all $s' \in b_I$ such that $s \neq s'$, we have that $\neg T(s')$ is true.

Atoms ϕ/t are achieved through axioms $\langle \phi/t, \bigwedge_{s' \in b_I}(\neg T(s') \vee \phi/s')\rangle$. Since for all $s' \in b_I$ such that $s \neq s'$, we have that $\neg T(s')$ is true, ϕ/t is true only if ϕ/s is true.

Lemma B.5. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves $D_i(s)$ in P', then $n(\pi)$ achieves $s \notin S_i(t)$ in P.*

Proof. Suppose the first action of π is $assume(s')$ and the hidden, true state in P is s^*. If π is empty, then $D_i(s)$ is true if $s \notin S_i(t)$.

If $\pi = \pi', \alpha$ achieves $D_i(s)$ in P', then (i) α after π' achieves $D_i(s)$, or (ii) π' achieves $D_i(s)$ and α does not delete it.

If (i), then α is (a) a sensing action **sense**$[i](\phi)$ or a passive sensor **psense**$[i](\phi)$, or (b) α is a physical action.

If (a) then α has a conditional effect $\phi/t \wedge \neg\phi/s \rightarrow D_i(s)$, then by Lemmas B.1 and B.4, ϕ is true in state s' of P and $\neg\phi$ is true in state s of P. Since $n(\pi)$ is applicable and consistent with the observations, state s^* and s' agree in the valuation of ϕ, therefore $n(\pi)$ achieves $s \notin S_i(t)$ in P. If (b) is true, then by Lemmas B.3 and B.4, $n(\pi)$ achieves $B(t), s' \models K_j Pre(\alpha)$ and $B(t), s \not\models K_j Pre(\alpha)$. Since $n(\pi)$ is applicable and consistent with the observations, $B(t), s' \models K_j Pre(\alpha)$, therefore $s \notin S_i(t)$ in P.

□

Lemma B.6. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves K_iL/t, where i the planning agent, in P', then $n(\pi)$ achieves $B(t), s_j \models L$ in P, for $s_j \in r_i(s^*, t)$, and s^* the hidden, true state.*

Proof: By previous lemma B.4, if K_iL/t is achieved and $assume(s)$ was the first action of π, then K_iL/s is achieved. K_iL/s is derived by the axiom $\langle K_iL/s, \bigwedge_{s' \in b_I}(L/s' \vee D_i(s, s'))$. By definition, $D_i(s')$ is true if and only if $D_i(s, s')$ is true, for $s' \in b_I$. This means that we can replace $D_i(s, s')$ with $D_i(s')$ in the definition of the axiom, which will become $\langle K_iL/s, \bigwedge_{s' \in b_I}(L/s' \vee D_i(s'))$. For K_iL/s to be true, L/s' must be true for all states s' such that $\neg D_i(s')$. Since $n(\pi)$ is consistent with the observations, $\neg D_i(s^*)$ must be true. Therefore, by lemmas B.1 to B.3, $B(t), s_j \models L$, for $s_j \in r_i(s^*, t)$.

Lemma B.7. *Suppose a planning agent i and a sequence of actions π that is applicable in P'. If $n(\pi)$ is consistent with the observations, then $n(\pi)$ is applicable P with s^* the hidden, true state.*

Proof. If π is empty it is trivial. Suppose $\pi = assume(s), \pi', a, \epsilon$. Since π is applicable in P', then π' is applicable in P', and by inductive hypothesis, $n(\pi)$ is applicable in P. Furthermore, since π is applicable in P'. then π' achieves $K_j Pre(\alpha)/t$, for action α and owner of the action agent j in P'. By Lemma B.4 and B.6, π achieves $K_j Pre(\alpha)/s$, and since $n(\pi)$ is consistent with observations, $n(\pi)$ achieves $B(t), s^* \models K_j Pre(\alpha)$, therefore, $n(\pi)$ is applicable in P. □

Theorem 3 (p.66) *If π is a sequence of actions that achieves G' in P', then $n(\pi)$ achieves the goal G in P for a hidden, true state s^*, if $n(\pi)$ is consistent with the observations, b) Otherwise, if π' is the shortest prefix of π that is inconsistent and π includes the action $assume(s)$, after the execution of $n(\pi')$ in P, $s \notin S_i(t')$ where t' is the resulting time step.*

Proof. Direct from previous lemma. Since π achieves G' then π, for all $s \in b_I$, achieves either G/s or $D_i(s)$. Since $n(\pi)$ is consistent with the observations, $D_i(s)$ (and, by extension, $s \notin S_i(t)$ in P) cannot become true, for $s^* = s$. Thus, since π achieves G/s for $s \in S_i(t)$, then $n(\pi)$ achieves $B(t), s \models G$. □

Lemma B.8. *Suppose a sequence of actions π, with first action $assume(s)$ that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If $n(\pi)$ achieves $s \models L$ in P, where L is an objective atom, then π achieves L/s in P'.*

Proof: Suppose π is empty. If $I \models s$ and $s \models L$ then $L/s \in I'$.

Suppose $\pi = \pi', \alpha$. There are two cases: (i) α achieves, after $n(\pi')$, L at state s, or (ii) $n(\pi')$ achieves L and α does not delete it.

If (i), α is a physical action with conditional effect $C \to L$, and $n(\pi')$ achieves C in state s in P. Then α in P' has a conditional effect $C/s \to L/s$, and by inductive hypothesis, π' must achieve C/s. Therefore, π achieves L/s. If (ii) is true, by inductive hypothesis, π' must achieve C/s and $\neg L'/s$ for some $L'/s \in C'/s$, so π achieves L/s.

Lemma B.9. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If $n(\pi)$ achieves $s' \notin r_i(s, t)$ in*

*P due to a sensing action **sense**$[i](\phi)$ or a passive sensor **psense**$[i](\phi)$, where ϕ is an objective formula, π achieves $D_i(s, s')$ in P'.*

Proof: Suppose $n(\pi) = \pi$ is empty. If $s' \notin r_i(s, t)$, then $D_i(s, s') \in I'$.

Suppose $n(\pi) = n(\pi'), \alpha$. There are two cases: (i) α achieves, after $n(\pi)$, $s \notin r_i(s, t)$, or (ii) $n(\pi)$ achieves $s \notin r_i(s, t)$ and α does not achieve $s \in r_i(s, t)$.

If (i), α is a sensing action **sense**$[i](\phi)$ in P, or a physical action followed by a passive sensor **psense**$[i](\phi)$ in P. Suppose α is a sensing action. Since α achieves $s' \notin r_i(s, t)$, $n(\pi')$ achieves $s \models \phi$ and $s' \not\models \phi$. By Lemma B.8, π' achieves ϕ/s and $\neg\phi/s'$. The sensing action $\alpha =$ **sense**$[i](\phi)$ in P' has a conditional effect $\phi/s \wedge \neg\phi/s' \to D_i(s, s')$. Therefore, $D_i(s, s')$ is achieved in P'. Similar, if $\alpha =$ **psense**$[i](\phi)$. if (ii), it is direct since there is no action that can add a state to a set $r_i(s, t)$.

Lemma B.10. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If $n(\pi)$ achieves $B(t), s \models K_i L$ in P, then π achieves $K_i L/s$ in P'.*

Proof: Since $n(\pi)$ achieves $B(t), s \models K_i L$, then for all states $s' \in r_i(s, t)$, we have $s' \models L$.

Epistemic literals $K_i L/s$ in P' are achieved through axioms of the form $\langle K_i L/s, \bigwedge_{s' \in b_I}(L/s' \vee D_i(s, s'))\rangle$.

Suppose that π does not achieve $K_i L/s$. It must be that there exists a state s'' such that $\neg L/s''$ and $\neg D_i(s, s'')$. By Lemma B.8 we have that if $n(\pi)$ achieves $s' \models L$, π achieves L/s'. By Lemma B.9 we have that if $s' \notin r_i(s, t)$ is true after the execution of $n(\pi)$, $D_i(s, s')$ is true in π. Therefore, $K_i L/s$ is achieved by π.

Lemma B.11. *Suppose a sequence of actions π that is applicable in P' and the first action in π is $assume(s)$. Further, suppose $n(\pi)$ is applicable and consistent with the observations in P, where the hidden true state is s^*. If $n(\pi)$ achieves $B(t), s^* \models K_i L$, then π achieves $K_i L/t$.*

Proof: Since $B(t), s^* \models K_i L$ is true, then $B(t), s' \models L$, for all $s' \in r_i(s^*, t)$. The definition of the axiom for a derived atom ϕ/t is $\bigwedge_{s' \in b_I}(\neg T(s') \vee \phi/s')$. This means that, since $assume(s)$ is unique and makes $T(s)$ true, $K_i L/t$ is true if $K_i L/s$ is true. The definition of the axiom $K_i L/s$ is $\bigwedge_{s' \in b_I}(L/s' \vee D_i(s, s'))$.

Suppose $K_i L/s$ is not achieved. This means that there exists a state s'' such that $\neg L/s''$ and $\neg D_i(s, s'')$. By Lemmas B.8 and B.9 we have that since $B(t), s' \models L$, for all $s' \in r_i(s^*, t)$, then for all states s' either L/s'' is true or $D_i(s, s'')$. Therefore, $K_i L/t$ is true.

Lemma B.12. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If $n(\pi)$ achieves $s \notin S_i(t)$ in P, then π achieves $D_i(s)$ in P'.*

Proof: Suppose the first action of π is $assume(s')$ and the hidden, true state in P is s^*, where $s^* \neq s'$.

If $n(\pi)$ is empty, since $s \notin S_i(t)$ in P, then $D_i(s) \in I'$ in P'.

Suppose $n(\pi) = n(\pi'), \alpha$. There are two cases: (i) α after $n(\pi')$ achieves $s \notin S_i(t)$, or (ii) $n(\pi')$ achieves $s \notin S_i(t)$ and α does not achieve $s \in S_i(t)$.

If (i) there are two more cases: (a) α is a sensing action or a passive sensor, or (b) α is a physical action.

If (a), then agent i senses the truth value of a formula ϕ, such that $B(t), s^* \models \phi$ and $B(t), s \not\models \phi$. Furthermore, since π is applicable and $n(\pi)$ consistent with the observations, the sensing action/passive sensor has a conditional effect $\phi/t \wedge \neg\phi/s \rightarrow D_i(s)$, and by lemmas B.8 and B.10, we have that π achieves $D_i(s)$. If (b) α with owner j is such that the precondition is $K_jPre(\alpha)$ and $B(t), s \not\models K_jPre(\alpha)$. Since π is applicable, and by Lemmas B.10 and B.11, we have that π achieves $D_i(s)$.

If (ii), it is direct since no action can achieve $s \in S_i(t)$.

Lemma B.13. *If an action sequence $n(\pi)$ is applicable in P, then $\pi = assume(s), \pi'$ is applicable in P', if π is consistent with the observations.*

Proof: If $n(\pi)$ is empty, it is trivial. Suppose $n(\pi) = n(\pi'), \alpha$ is applicable in P. Since α is applied, its preconditions are achieved by $n(\pi')$, and by inductive hypothesis, $n(\pi')$ is applicable in P. Also, since $n(\pi')$ achieves $B(t), s \models K_jPre(\alpha)$, for $s \in S_j(t)$, by Lemma B.10 we have that π' achieves $K_jPre(\alpha)/s$ for the same states. By Lemma B.11, π achieves the preconditions of the actions and, thus, is applicable in P', if consistent with observations.

Theorem 4 (p.66) *Suppose an action sequence $\pi = assumes(s), \pi'$ and the corresponding normalized sequence $n(\pi)$. If $n(\pi)$ is a plan that achieves K_iG for P, for a hidden true state s^*, then π is a plan for P' for $s = s^*$, and any such plan is consistent.*

Proof. Direct from previous lemma, if we consider a problem P'' same as P, with the addition of an action α' which has as a precondition K_iG. If $n(\pi)$ is a plan for P, then $\pi' = \pi, a'$ is a plan for P'', and by the previous lemma π' is applicable in P', thus it achieves K_iG. Furthermore, any such plan is consistent since, by previous theorem, the shortest prefix of the plan that is consistent and applied is sound, thus, if $s \models \phi$ in P, ϕ/s is true. Therefore, the observation cannot be inconsistent. $\square$

Theorem 5 (p.66) *If the executions in P cannot reach a dead-end, Algorithm 1 will solve P after a number of calls to the classical planner that is bounded by $|b_I| \times |A|^2$, where b_I is the set of initial states in P and A is the set of agents.*

Proof. Direct since the translation is sound and complete and the protocol chooses a new planning agent j as long as $S_j(t) \neq \emptyset$. At every iteration, either the goal is achieved due to previous theorems, or a state s is removed from $S_i(t)$, where i the current planning agent. Thus, in the worst case, there will be $|b_I| \times |A|$ calls to the planner.

$\square$

APPENDIX **C**

Properties of $K(P, B(t), S_i(t), O^+, O(t))$ translation

From now on we will refer to the translation $K(P, B(t), S_i(t), O^+(t), O(t))$ as P'.

Lemma C.1. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves L/s in P', then $n(\pi)$ achieves $s \models L$ in P.*

Proof. Suppose π, and by extension $n(\pi)$, is empty. Since L/s is true in I', then $I \models s$ and $s \models L$.

Suppose $\pi = \pi', \alpha$. There are two cases: (i) α achieves L/s after π', or (ii) π' achieves L/s and α does not delete it.

If (i) is true, α in P' contains a conditional effect $C/s \rightarrow L/s$, and π' achieved C/s in P'. Also, α in P must contain a conditional effect $C \rightarrow L$, and by inductive hypothesis, $n(\pi')$ must achieve C in state s of P. Therefore, $n(\pi)$ achieves L in state s of P. If (ii) is true, π achieves L/s and α is a physical action with a conditional effect $C'/s \rightarrow \neg L/s$, and by inductive hypothesis, $n(\pi')$ achieves L and $\neg L'$ in state s of P, so $n(\pi)$ achieves L is state s of P. $\square$

The $K(P, B(t), S_i(t), O^+(t), O(t))$ translation introduces atoms $D(s)$, $D_i(s)$ and $a^+_{A_k,\phi}$: the first represents that the state s is completely removed from the set of possible states (and that is common knowledge), the second that the planning agent i does not consider the state s possible, and the third that the sensing action corresponding to the specific auxiliary atom has been applied. These atoms, after they have been achieved, cannot become false. For $D(s)$ and $D_i(s)$, if the states have been deemed not possible at some time t due to an observation that is inconsistent with the hidden state, they cannot become possible at a later time $t' > t$. For $a^+_{A_k,\phi}$, similarly, if the action has been applied then the agents in A_k can distinguish between certain states s and s' due to that action and, since all actions are public and deterministic, they can distinguish the states from that point on.

Lemma C.2. *Suppose a sequence of actions π that is applicable in P' and the first action in π is $assume(s)$. We have that ϕ/t is achieved if and only if ϕ/s is achieved.*

Proof. Since there is only one $assume(s)$ action that appears in π, $s \in S_i(t)$, as the first action, and its effect is $T(s)$, then for all $s' \in b_I$ such that $s \neq s'$, we have that $\neg T(s')$ is true.

Atoms ϕ/t are achieved through axioms $\langle \phi/t, \bigwedge_{s' \in b_I}(\neg T(s') \vee \phi/s') \rangle$. Since for all $s' \in b_I$ such that $s \neq s'$, we have that $\neg T(s')$ is true, ϕ/t is true only if ϕ/s is true. □

Lemma C.3. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves $D(s')$ in P' due to a sensing action **sense**$[A](\phi)$ or a passive sensor **psense**$[A](\phi)$, where ϕ is an objective formula, then $n(\pi)$ achieves $s' \notin r_i(s,t)$ in P.*

Proof. Suppose π is empty. Since $D(s') \in I'$ then $s' \notin r_i(s,t)$.

Suppose $\pi = \pi', \alpha$. There are two cases to consider: (i) action α after π' achieves $D(s')$ ($D_i(s')$), or (ii) π' achieves $D(s')$ and action α does not delete it.

If (i) is true, then α has a conditional effect $\phi/t \wedge \neg\phi/s' \rightarrow D(s')$. By Lemma $C2$, since ϕ/t is achieved, ϕ/s is achieved. By Lemma $C1$, $n(\pi')$ achieves $s \models \phi$ and $s' \not\models \phi$. Therefore, by definition of the sensing action, we have $s' \notin r_i(s,t)$. If (ii) is true, it is direct since $D(s')$ cannot be deleted. □

Lemma C.4. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves $\alpha^+_{A_k,\phi}$ in P', such that $\alpha^+_{A_k,\phi} \in O(s,s',i)$ and $O(s,s',i) \in O^+$, then π achieves $s' \notin r_i(s,t)$.*

Proof. By definition of the construction of $O(s,s',i)$. Since $alpha^+_{A_k,\phi} \in O(s,s',i)$, then ϕ is a static formula and the states s and s' disagree on its truth value. Since π achieves $\alpha^+_{A_k,\phi}$, there was a sensing action or a sensor which allowed agent $i \in A_k$ to sense the truth value of ϕ, and thus, distinguish between states s and s' ($s' \notin r_i(s,t)$). $\square$

Lemma C.5. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves K_iL/s in P', then $n(\pi)$ achieves $B(t), s \models K_iL$ in P.*

Proof. Epistemic literals K_iL/s are achieved through axioms of the form $\langle K_iL/s, \wedge_{s'\in b_I}[L/s' \vee D(s') \bigvee_{a[A_k])(\phi)\in O^+(s,s',i)} a^+_{A_k,\phi}]\rangle$, for $O^+(s,s',i) \in O^+$. $\square$

Proof: Epistemic literals K_iL/s are achieved through axioms of the form $\langle K_iL/s, \bigwedge_{s'\in b_I}(L/s' \vee D_i(s,s'))\rangle$.

Since π achieved K_iL/s, for all states $s' \in b_I$, either L/s' is achieved or $D(s')$ is achieved, or an atom $a^+_{A_k,\phi}$ which denotes that agent i sensed some formula ϕ which allows him to distinguish between s and s'.

Suppose that $B(t), s \not\models K_iL$. It must be that there exists a state $s' \in b_I$ such that (i) $s' \not\models L$ and (ii) $s' \in r_i(s,t)$. By Lemma C1 we have that if π achieves L/s', then $n(\pi)$ achieves $s \models L$ in P. By Lemma C3 we have that if π achieves $D(s')$, then $n(\pi)$ achieves $s' \notin r_i(s,t)$ in P. By Lemma C4 we have that if π achieves $a^+_{A_k,\phi}$, and $a^+_{A_k,\phi} \in O(s,s',i)$, then $n(\pi)$ achieves $s' \notin r_i(s,t)$ in P.

Since for all states we have that $s \models L$ is achieved or $s \notin r_i(s,t)$ is achieved, there cannot be a state s' s.t. $s' \not\models L$ and $s' \in r_i(s,t)$. Therefore, $n(\pi)$ must achieve $B(t), s \models K_iL$.

Lemma C.6. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves K_iL/t, where i the planning agent, in P', then $n(\pi)$ achieves $B(t), s_j \models L$ in P, for $s_j \in r_i(s^*,t)$, and s^* the hidden, true state.*

Proof: By lemmas C2 and C5, if K_iL/t is achieved and $assume(s)$ was the first action of π, then K_iL/s is achieved.

K_iL/s is derived by the axiom $\langle K_iL/s, \bigwedge_{s'\in b_I}(L/s' \vee D_i(s,s'))$. By definition, $D_i(s')$ is true if and only if $D_i(s,s')$ is true, for $s' \in b_I$. This means that we can replace $D_i(s,s')$ with $D_i(s')$ in the definition of the axiom, which will become $\langle K_iL/s, \bigwedge_{s'\in b_I}(L/s' \vee D_i(s'))$. For K_iL/s to be true, L/s' must be true for all states s' such that $\neg D_i(s')$. Since $n(\pi)$ is consistent with the observations, $\neg D_i(s^*)$ must be true. Therefore, by lemmas B.0.1 to B.0.3, $B(t), s_j \models L$, for $s_j \in r_i(s^*,t)$.

Lemma C.7. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If π achieves $D_i(s)$ in P', then $n(\pi)$ achieves $s \notin S_i(t)$ in P.*

Proof. Suppose the first action of π is $assume(s')$ and the hidden, true state in P is s^*. If π is empty, then $D_i(s)$ is true if $s \notin S_i(t)$.

Suppose $\pi = \pi', \alpha, \epsilon$. There are two cases to consider: (i) action α after π' achieves $D_i(s)$ in P', or (ii) π' achieves $D_i(s)$ and α does not delete it.

If (i) is true, then α must be a sensing action **sense**$[i](\phi)$, or ϵ contains a passive sensor **psense**$[i](\phi)$, each with conditional effect $\phi/t \wedge \neg\phi/s \rightarrow D_i(s)$. By Lemmas C1 and C2, $n(\pi')$ achieves $s' \models \phi$ and $s \not\models \phi$. Since $n(\pi)$ is consistent with observations and the ϕ is true in assumed state s', we have that $s^* \models \phi$ as well. Therefore, $n(\pi)$ achieves $s \notin S_i(t)$. if (ii) is true, it is direct by inductive hypothesis and the fact that $D_i(s)$ cannot be deleted by an action.

□

Lemma C.8. *Suppose a planning agent i and a sequence of actions π that is applicable in P'. If $n(\pi)$ is consistent with the observations, then $n(\pi)$ is applicable P with s^* the hidden, true state.*

Proof. If π is empty it is trivial. Suppose $\pi = assume(s), \pi', a, \epsilon$. Since π is applicable in P', then π' is applicable in P', and by inductive hypothesis, $n(\pi)$ is applicable in P. Furthermore, since π is applicable in P'. then π' achieves $K_jPre(\alpha)/t$, for action α and owner of the action agent j in P'. By Lemmas C2 and C5, π achieves $K_jPre(\alpha)/s$, and since $n(\pi)$ is consistent with observations, $n(\pi)$ achieves $B(t), s^* \models K_jPre(\alpha)$, therefore, $n(\pi)$ is applicable in P. □

Theorem C.1. *If π is a sequence of actions that achieves G' in P', then $n(\pi)$ achieves the goal G in P for a hidden, true state s^*, if $n(\pi)$ is consistent with the observations.*

Proof: Direct from previous lemma. Since π achieves G' then π, for all $s \in b_I$, achieves either G/s or $D_i(s)$. Since $n(\pi)$ is consistent with the observations then $D_i(s)$ (and, by extension, $s \notin S_i(t)$ in P) cannot become true, for $s^* = s$. Thus, since π achieves G/s for $s \in S_i(t)$, then $n(\pi)$ achieves $B(t), s \models G$.

Lemma C.9. *Suppose a sequence of actions π, with first action $assume(s)$ that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If $n(\pi)$ achieves $s \models L$ in P, where L is an objective atom, then π achieves L/s in P'.*

Proof. Suppose π is empty. If $I \models s$ and $s \models L$ then $L/s \in I'$.

Suppose $\pi = \pi', \alpha, \epsilon$, There are two cases to consider: (i) α achieves $s \models L$, or (ii) π' achieves $s \models L$ and α does not achieve $s \not{models} L$.

If (i) is true, then α in P' has a conditional effect $C/s \to L/s$, and by inductive hypothesis, π' must achieve C/s. Therefore, π achieves L/s. If (ii) is true, by inductive hypothesis, π' must achieve C/s and $\neg L'/s$ for some $L'/s \in C'/s$, so π achieves L/s. □

Lemma C.10. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If $n(\pi)$ achieves $s' \notin r_i(s,t)$ in P due to a sensing action **sense**$[B](\phi)$ or a passive sensor **psense**$[B](\phi)$, where ϕ is an objective formula, π achieves $D(s)$ or $a^+_{B,\phi}$ in P'.*

Proof: Suppose $n(\pi) = \pi$ is empty. There are two cases: (i) $s' \notin r_i(s,t)$ is true for $B = A$ and $s \in b_I$, or (ii) $s' \notin r_i(s,t)$ is true for some $s \in b_I$.

If (i) is true, then $I' \models D(s)$. If (ii) is true, then $s' \notin r_i(s,t)$ can only be achieved by the application of a sensing action or a passive sensor during some earlier execution, with ϕ being a static formula. Therefore, $I' \models a^+_{B,\phi}$ due to $a^+_{B,\phi} \in O(t)$

Suppose $n(\pi) = n(\pi'), \alpha$. There are two cases to consider: (i) α after $n(pi')$ achieved $s' \notin r_i(s,t)$, or (ii) $n(pi')$ achieved $s' \notin r_i(s,t)$ and α did not reverse it.

If (i) is true, and $B = A$, α achieves $s' \notin r_i(s,t)$ and since π is applicable, π achieves $a^+_{B,\phi}$. If (ii), it is direct since there is no action that adds a state in a set $r_i(s,t)$.

Lemma C.11. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If $n(\pi)$ achieves $B(t), s \models K_iL$ in P, then π achieves K_iL/s in P'.*

Proof: Since $n(\pi)$ achieves $B(t), s \models K_iL$, then for all states $s' \in r_i(s,t)$, we have $s' \models L$.

Epistemic literals K_iL/s in P' are achieved through axioms of the form $\langle K_iL/s, \wedge_{s' \in b_I}$ $[L/s' \vee D(s') \bigvee_{a[A_k])(\phi) \in O^+(s,s',i)} a^+_{A_k,\phi}]\rangle$, for $O^+(s,s',i) \in O^+$.

Suppose that $n(\pi)$ achieves $B(t), s \models K_iL$ but π does not achieve K_iL/s. It must be that there exists a state s' such that $\neg L/s'$ and agent i cannot distinguish between s and s', which entails that $\neg D(s')$ is true and $O^+(s,s',i)$ is empty. By Lemma C9 we have that if $n(\pi)$ achieves $s' \models L$, π achieves L/s'. By Lemma C10 we have that if $n(\pi)$ achieves $s' \notin r_i(s,t)$, then (a) it was the effect of an action **sense**$[A](\phi)$ and $D(s)$ is achieved, or (b) it was the effect of an action **sense**$[B](\phi)$, with $B \subset A$, thus ϕ is static and $a^+_{B,\phi} \in O(s, s', i$. Therefore, and since π is applicable in P', the action is applied and achieves $a^+_{B,\phi}$. Therefore, K_iL/s is achieved by π.

Lemma C.12. *Suppose a sequence of actions π that is applicable in P' and the first action in π is $assume(s)$. Further, suppose $n(\pi)$ is applicable and consistent with the observations in P, where the hidden true state is s^*. If $n(\pi)$ achieves $B(t), s^* \models K_iL$, then π achieves K_iL/t.*

Proof: Since $B(t), s^* \models K_iL$ is true, then $B(t), s' \models L$, for all $s' \in r_i(s^*, t)$. The definition of the axiom for a derived atom ϕ/t is $\bigwedge_{s' \in b_I}(\neg T(s') \vee \phi/s')$. This means that, since $assume(s)$ is unique and makes $T(s)$ true, K_iL/t is true if K_iL/s is true. The definition of the axiom K_iL/s is $\bigwedge_{s' \in b_I}(L/s' \vee D_i(s, s'))$.

Suppose K_iL/s is not achieved. This means that there exists a state s'' such that $\neg L/s''$ and $\neg D_i(s, s'')$. By Lemmas C1 and C3 we have that since $B(t), s' \models L$, for all $s' \in r_i(s^*, t)$ we have $s' \models L$, which is a contradiction. Therefore, K_iL/t is true.

Lemma C.13. *Suppose a sequence of actions π that is applicable in P' and $n(\pi)$ that is applicable and consistent with the observations in P. If $n(\pi)$ achieves $s \notin S_i(t)$ in P, then π achieves $D_i(s)$ in P'.*

Proof: Suppose the first action of $\pi = \pi', \alpha$ is $assume(s')$ and the hidden, true state in P is s^*, where $s^* \neq s'$.

There are two cases to consider: (i) α, after π achieves $s \notin S_i(t)$, or (ii) π achieves $s \notin S_i(t)$ and α does not add the state back in $S_i(t)$.

If (i), then (a) α is a sensing action or a passive sensor, or (b) α is a physical action with owner j and precondition $K_jPre(\alpha)$.

If (a), agent i senses the truth value of a formula ϕ, such that $B(t), s^* \models \phi$ and $B(t), s \not\models \phi$. Since π is applicable and $n(\pi)$ consistent with the observations, the sensing action/passive sensor has a conditional effect $\phi/t \wedge \neg\phi/s \rightarrow D_i(s)$, by lemmas C2 to C4, we have that π achieves $D_i(s)$.

If (b) α with owner j has precondition $K_jPre(\alpha)$, and $B(t), s \not\models K_jPre(\alpha)$. By Lemmas C2 and C5, we have that π achieves $D_i(s)$.

Lemma C.14. *If an action sequence $n(\pi)$ is applicable in P, then $\pi = assume(s), \pi'$ is applicable in P', if π is consistent with the observations.*

Proof: If $n(\pi)$ is empty, it is trivial. Suppose $n(\pi) = n(\pi'), \alpha$ is applicable in P. Since α is applied, its preconditions are achieved by $n(\pi')$, and by inductive hypothesis, $n(\pi')$ is applicable in P. Also, since $n(\pi')$ achieves $B(t), s \models K_jPre(\alpha)$, for $s \in S_j(t)$, by

Lemma B.0.10 we have that π' achieves $K_jPre(\alpha)/s$ for the same states. Therefore, π is applicable in P', if consistent with observations.

Theorem C.2. *Suppose an action sequence $\pi = assumes(s), \pi'$ and the corresponding normalized sequence $n(\pi)$. If $n(\pi)$ is a plan that achieves K_iG for P, for a hidden true state s^*, then π is a plan for P' for $s = s^*$, and any such plan is consistent.*

Proof. Direct from previous theorem, if we consider a problem P'' same as P, with the addition of an action α' which has as a precondition K_iG. If $n(\pi)$ is a plan for P, then $\pi' = \pi, a'$ is a plan for P'', and by the previous lemma π' is applicable in P', thus it achieves K_iG. Furthermore, any such plan is consistent since, by previous theorem, the shortest prefix of the plan that is consistent and applied is sound, thus, if $s \models \phi$ in P, ϕ/s is true. Therefore, the observation cannot be inconsistent. $\square$

APPENDIX **D**

Properties of $K(P, B_D(t), S_i^D(t))$ translation

The proofs for the $K(P, B_D(t), S_i^D(t))$ translation are exactly the same with $K(P, B(t), S_i(t))$ if, instead of considering the joint belief $B(t)$, we consider the partial belief $B_j(t) \in B_D(t)$ that corresponds to every sensing action or conditional effect. Thus, literals of the form L/s will be written as $L/(s, j)$, $D_i(s)$ as $D_i(s, j)$, $r_i(s, t)$ as $r_i^j(s, t)$ etc.

Bibliography

Bibliography

Albore, A., Palacios, H., and Geffner, H. (2009). A translation-based approach to contingent planning. In *Proc. IJCAI-09*, pages 1623–1628.

Albore, A., Palacios, H., and Geffner, H. (2010). Compiling uncertainty away in non-deterministic conformant planning. In *Proc. ECAI*, pages 465–470.

Aucher, G. and Bolander, T. (2013). Undecidability in epistemic planning. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 27–33. AAAI Press.

Aucher, G. and Schwarzentruber, F. (2013). On the complexity of dynamic epistemic logic.

Baltag, A. (2000). A logic for suspicious players: Epistemic actions and belief-update in games. *Report-Software engineering*, (44):1–30.

Baltag, A. and Moss, L. S. (2004a). Logics for epistemic programs. *Synthese*, 139(2):165–224.

Baltag, A. and Moss, L. S. (2004b). Logics for epistemic programs. *Synthese*, 139(2):165–224.

Baltag, A., Moss, L. S., and Solecki, S. (1998). The logic of public announcements, common knowledge, and private suspicions. In *Proc. of the 7th Conf. on Theoretical aspects of rationality and knowledge*, pages 43–56.

Baral, C., Gelfond, G., Pontelli, E., and Son, T. C. (2012). An action language for reasoning about beliefs in multi-agent domains. In *Proc. of the 14th International Workshop on Non-Monotonic Reasoning*.

Bernstein, D., Zilberstein, S., and Immerman, N. (2000). The complexity of decentralized control of Markov decision processes. In *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence*, pages 32–37.

Bertoli, P., Cimatti, A., Roveri, M., and Traverso, P. (2001). Planning in nondeterministic domains under partial observability via symbolic model checking. In *Proc. IJCAI-01*.

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.

Bonet, B. and Geffner, H. (1999). Planning as heuristic search: New results. In *European Conference on Planning*, pages 360–372. Springer.

Bonet, B. and Geffner, H. (2000). Planning with incomplete information as heuristic search in belief space. In *Proc. of AIPS-2000*, pages 52–61.

Bonet, B. and Geffner, H. (2001). Planning as heuristic search. *Artificial Intelligence*, 129(1–2):5–33.

Bonet, B. and Geffner, H. (2008). Heuristics for planning with penalties and rewards formulated in logic and computed through circuits. *Artificial Intelligence*, 172(12-13):1579–1604.

Bonet, B. and Geffner, H. (2014a). Belief tracking for planning with sensing: Width, complexity and approximations. *Journal of Artificial Intelligence Research*, 50:923–970.

Bonet, B. and Geffner, H. (2014b). Flexible and scalable partially observable planning with linear translations. In *Proc. AAAI*, pages 2235–2241.

Bonet, B., Loerincs, G., and Geffner, H. (1997). A robust and fast action selection mechanism for planning. In *Proc. AAAI-97*, pages 714–719.

Brafman, R. and Hoffmann, J. (2004). Conformant planning via heuristic forward search: A new approach. In *Proc. ICAPS-04*.

Brafman, R. I. and Beer Sheva, I. (2015). A privacy preserving algorithm for multi-agent planning and search. In *IJCAI*, pages 1530–1536.

Brafman, R. I. and Domshlak, C. (2008). From one to many: Planning for loosely coupled multi-agent systems. In *ICAPS*, pages 28–35.

Brafman, R. I. and Shani, G. (2012a). A multi-path compilation approach to contingent planning. In *Proc. AAAI*.

Brafman, R. I. and Shani, G. (2012b). Replanning in domains with partial information and sensing actions. *Journal of Artificial Intelligence Research*, 45(1):565–600.

Brafman, R. I., Shani, G., and Zilberstein, S. (2013). Qualitative planning under partial observability in multi-agent domains. In *Proc. AAAI*.

Brenner, M. (2010). Creating dynamic story plots with continual multiagent planning. In *Proc. AAAI*.

Bylander, T. (1994). The computational complexity of STRIPS planning. *Artificial Intelligence*, 69:165–204.

Cooper, M., Herzig, A., Maffre, F., Maris, F., and Regnier, P. (2016). A simple account of multiagent epistemic planning. In *Proc. ECAI*.

Darwiche, A. and Marquis, P. (2002). A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264.

De Weerdt, M. and Clement, B. (2009). Introduction to planning in multiagent systems. *Multiagent and Grid Systems*, 5(4):345–355.

Engesser, T., Bolander, T., Mattmüller, R., and Nebel, B. (2015). Cooperative epistemic multi-agent planning with implicit coordination. In *Proc. Workshop on Distributed and Multi-Agent Planning (DMAP-15)*, pages 68–76.

Entringer, R. C. and Slater, P. J. (1979). Gossips and telegraphs. *Journal of the Franklin Institute*, 307(6):353–360.

Fagin, R., Halpern, J., Moses, Y., and Vardi, M. (1995). *Reasoning about Knowledge*. MIT Press.

Fikes, R. and Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 1:27–120.

Fox, M. and Long, D. (2003). PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of AI Research*, 20.

Geffner, H. and Bonet, B. (2013). *A Concise Introduction to Models and Methods for Automated Planning*. Morgan & Claypool Publishers.

Gerbrandy, J. and Groeneveld, W. (1997). Reasoning about information change. *Journal of logic, language and information*, 6(2):147–169.

Goldman, C. V. and Zilberstein, S. (2003). Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 137–144. ACM.

Gomes, C. P., Kautz, H., Sabharwal, A., and Selman, B. (2008). Satisfiability solvers. *Foundations of Artificial Intelligence*, 3:89–134.

Halpern, J. Y. and Lakemeyer, G. (2001). Multi-agent only knowing. *Journal of Logic and Computation*, 11(1):41–70.

Harary, F. and Schwenk, A. J. (1974). The communication problem on graphs and digraphs. *Journal of the Franklin Institute*, 297(6):491–495.

Haslum, P. and Jonsson, P. (1999). Some results on the complexity of planning with incomplete information. In *Proc. ECP-99, Lect. Notes in AI Vol 1809*, pages 308–318. Springer.

Hedetniemi, S. M., Hedetniemi, S. T., and Liestman, A. L. (1988). A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349.

Helmert, M. (2006). The Fast Downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246.

Helmert, M. and Domshlak, C. (2009). Landmarks, critical paths and abstractions: what's the difference anyway? In *ICAPS*, pages 162–169.

Herzig, A., Lang, J., and Marquis, P. (2005). Action progression and revision in multiagent belief structures. In *Proc. 6th Workshop on Nonmonotonic Reasoning, Action, and Change (NRAC 2005)*.

Herzig, A. and Maffre, F. (2017). How to share knowledge by gossiping. *AI Communications*, 30(1):1–17.

Hintikka, J. (1962). Knowledge and belief.

Hoffmann, J. and Nebel, B. (2001). The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302.

Johnson, M., Jonker, C., van Riemsdijk, B., Feltovich, P., and Bradshaw, J. M. (2009). Joint activity testbed: Blocks world for teams (bw4t). In *Engineering Societies in the Agents World X*, pages 254–256. Springer.

Jonsson, A. and Rovatsos, M. (2011). Scaling up multiagent planning: A best-response approach.

Kautz, H. and Selman, B. (1996). Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proc. AAAI*, pages 1194–1201.

Kominis, F. and Geffner, H. (2015). Beliefs in multiagent planning: From one agent to many. In *Proc. ICAPS*, pages 147–155.

Kooi, B. and van Benthem, J. (2004). Reduction axioms for epistemic actions. *AiML-2004: Advances in Modal Logic, number UMCS-04-9-1 in Technical Report Series*, pages 197–211.

Kraus, S. and Lehmann, D. (1988). Knowledge, belief and time. *Theoretical Computer Science*, 58(1-3):155–174.

Kripke, S. (1971). Semantical considerations on modal logic. In Linsky, L., editor, *Reference and Modality*, pages 63–72. Oxford University Press.

Kripke, S. A. (1963). Semantical analysis of modal logic i normal modal propositional calculi. *Mathematical Logic Quarterly*, 9(5-6):67–96.

Levesque, H. (1990). All I know: a study in autoepistemic logic. *Artificial intelligence*, 42(2):263–309.

Lipovetzky, N. and Geffner, H. (2012). Width and serialization of classical planning problems. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 540–545. IOS Press.

Lipovetzky, N. and Geffner, H. (2017). Best-first width search: Exploration and exploitation in classical planning. In *AAAI*, pages 3590–3596.

Löwe, B., Pacuit, E., and Witzel, A. (2011). DEL planning and some tractable cases. In *Logic, Rationality, and Interaction*, pages 179–192. Springer.

Lutz, C. (2006). Complexity and succinctness of public announcement logic. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 137–143. ACM.

McDermott, D. (1996). A heuristic estimator for means-ends analysis in planning. In *Proc. AIPS-96*, pages 142–149.

McDermott, D. (2000). The 1998 AI Planning Systems Competition. *Artificial Intelligence Magazine*, 21(2):35–56.

McDermott, D. (2003). The formal semantics of processes in pddl. In *Proc. ICAPS-03 Workshop on PDDL*, pages 87–94.

Muise, C., Belle, V., Felli, P., McIlraith, S., Miller, T., Pearce, A. R., and Sonenberg, L. (2015). Planning over multi-agent epistemic states: A classical planning approach. In *Proc. AAAI*.

Palacios, H. and Geffner, H. (2006). Compiling uncertainty away: Solving conformant planning problems using a classical planner (sometimes). In *Proc. AAAI-06*.

Palacios, H. and Geffner, H. (2007). From conformant into classical planning: Efficient translations that may be complete too. In *Proc. 17th Int. Conf. on Planning and Scheduling (ICAPS-07)*.

Palacios, H. and Geffner, H. (2009). Compiling Uncertainty Away in Conformant Planning Problems with Bounded Width. *Journal of Artificial Intelligence Research*, 35:623–675.

Petrick, R. and Bacchus, F. (2002). A knowledge-based approach to planning with incomplete information and sensing. In *Proc. AIPS*, pages 212–222.

Richter, S. and Westphal, M. (2010). The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39(1):127–177.

Rintanen, J. (2004). Complexity of planning with partial observability. In *Proc. ICAPS-2004*, pages 345–354.

Shamma, J. (2008). *Cooperative control of distributed multi-agent systems*. John Wiley & Sons.

Son, T. C., Pontelli, E., Baral, C., and Gelfond, G. (2014). Finitary S5-theories. In *Logics in Artificial Intelligence*, pages 239–252. Springer.

Stone, P. (2002). Multiagent competitions and research: Lessons from robocup and tac. In *Robot Soccer World Cup*, pages 224–237. Springer.

Thiébaux, S., Hoffmann, J., and Nebel, B. (2005). In defense of pddl axioms. *Artif. Intell.*, 168(1-2):38–69.

To, S. T., Pontelli, E., and Son, T. C. (2011). On the effectiveness of CNF and DNF representations in contingent planning. In *Proc. IJCAI*, pages 2033–2038.

Tonino, H., Bos, A., de Weerdt, M., and Witteveen, C. (2002). Plan coordination by revision in collective agent based systems. *Artificial Intelligence*, 142(2):121–145.

Turner, H. (2002). Polynomial-length planning spans the polynomial hierarchy. In *JELIA '02: Proc. of the European Conference on Logics in AI*, pages 111–124. Springer-Verlag.

Vallati, M., Chrpa, L., and Kitchin, D. (2015). Portfolio-based planning: State of the art, common practice and open challenges. *AI Communications*, 28(4):717–733.

Van Benthem, J. (2011). *Logical dynamics of information and interaction*. Cambridge University Press.

Van Benthem, J., van Benthem, J. F., van Benthem, J. F., Mathématicien, I., and van Benthem, J. F. (2010). Modal logic for open minds.

Van Benthem, J., Van Eijck, J., and Kooi, B. (2006). Logics of communication and change. *Information and computation*, 204(11):1620–1662.

van der Hoek, W. (1990). Systems for knowledge and beliefs. In *European Workshop on Logics in Artificial Intelligence*, pages 267–281. Springer.

van Ditmarsch, H. and Kooi, B. (2008a). Semantic results for ontic and epistemic change. *Logic and the foundations of game and decision theory (LOFT 7)*, 3:87–117.

van Ditmarsch, H. and Kooi, B. (2008b). Semantic results for ontic and epistemic change. *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, pages 87–117.

van Ditmarsch, H., Ruan, J., and Verbrugge, R. (2008). Sum and product in dynamic epistemic logic. *Journal of Logic and Computation*, 18(4):563–588.

Van Ditmarsch, H., van Der Hoek, W., and Kooi, B. (2007). *Dynamic epistemic logic*, volume 337. Springer Science & Business Media.

van Ditmarsch, H., van der Hoek, W., and Kooi, B. (2007a). *Dynamic Epistemic Logic*. Springer.

van Ditmarsch, H., van der Hoek, W., and Kooi, B. (2007b). Dynamic epistemic logic and knowledge puzzles. In *Conceptual Structures: Knowledge Architectures for Smart Applications*, pages 45–58. Springer.

van Ditmarsch, H. P., van der Hoek, W., and Kooi, B. P. (2005). Dynamic epistemic logic with assignment. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 141–148. ACM.

Weiss, G. (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press.