Using Multiple Models of Reality:

On Agents who Know how to Play Safer

Wojciech Jamroga

Ph.D. dissertation committee:

Prof.dr. ir. A. Nijholt (University of Twente), promotor
Prof.dr. W. van der Hoek (University of Liverpool), co-promotor
Dr. J. Zwiers (University of Twente), assistant promotor
Dr. M.A. Bednarczyk (Polish Academy of Science)
Prof.dr. H. Brinksma (University of Twente)
Prof.dr. V. Goranko (Rand Afrikaans University)
Prof.dr. T. Huibers (University of Twente)
Prof.dr. J.-J.Ch. Meyer (Utrecht University)
Prof.dr. W.H.M. Zijm (University of Twente), chairman

Taaluitgeverij Neslia Paniculata.
Uitgeverij voor Lezers en Schrijvers van Talige Boeken,
Nieuwe Schoolweg 28, 7514 CG Enschede

CTIT Ph.D. Thesis series No. 04–64
Center for Telematics and Information Technology
P.O. Box 217, 7500 AE Enschede, The Netherlands

SIKS Dissertation Series No. 2004–13
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN 90–75296–11–8

ISSN 1381–3617 (CTIT Ph.D. Thesis No. 04–64)

Printed by: Acten, ul. Parkowa 4, Wejherowo, Poland; tel. +48 58 672 48 20.

Cover design: Wojtek Jamroga.
Based on ”Keep secrets of the State – the enemy may want to steal them right from you”, a Polish poster from the communist paranoia period, late 1940s.

USING MULTIPLE MODELS OF REALITY

ON AGENTS WHO KNOW HOW TO PLAY SAFER

DISSERTATION

to obtain
the doctor's degree at the University of Twente,
on the authority of the rector magnificus,
prof.dr. F.A. van Vught,
on account of the decision of the graduation committee,
to be publicly defended
on Thursday, July 1st, 2004, at 13.30.

by

Wojciech Janusz Jamroga

born on March 23rd, 1971

in Gdańsk, Poland

This dissertation is approved by:

prof. dr. ir. A. Nijholt (promotor)
prof. dr. W. van der Hoek (co-promotor)
Dr. J. Zwiers (assistant promotor)

Acknowledgments

First of all, I would like to thank Iwonka, my wife. For being together, sharing life with me – and for heroic listening to my ruminations about alternating-time logics in the middle of the night. To my children, Filipek and Ala, for asking questions when I read, hanging on my head when I write, and crying when I try to sleep. Sharing life in their own special, most intensive way, that is. Making me see everything anew with the eyes of a child. To the new one, approaching, hidden still on the other side of the skin, for its silent presence. To Krysia, my mum: for everything. To my friends: for being close to me no matter where we actually are; for trips and unexpected visits, late night discussions, and having a glass of beer... or wine... or tea... somewhere, from time to time. To God – for letting me be with my wife, my children, my family and my friends, and for the importance of everything that anybody does, says or thinks.

I would like to thank Andrzej W. Mostowski, my first boss at the University of Gdańsk, for encouragement and advice. To my colleagues back at the University, for all our defeatist discussions on impossibility of successful research. To Anton Nijholt at the University of Twente, for accepting me into his group under the ever-changing name (the name is Human Media Interaction Group right now; who knows what it will be like tomorrow?). To my supervisors in Twente: Anton Nijholt (again), Mannes Poel and Job Zwiers, for all the questions, answers and discussions. To other colleagues from the group: Bui The Duy, Jan Kuper, Martijn van Otterloo, Dennis Reidsma, Natasa Jovanovic, Rutger Rienks, Betsy van Dijk, Eelco Herder, Charlotte Bijron, Rieks op den Akker, Peter Asveld, Dirk Heylen, Andrei Malchanau, Hendri Hondorp, Alice Vissers, Mariet Theune, Lynn Packwood and Paul van der Vet: for talks, discussions, and permanent agent-to-agent interaction. To Rutger and Dennis (again) for their Dutch translation of the summary; it looks fantastic, and I am sure it reads equally well. It is a pity I cannot really appreciate this myself with my handicapped Dutch.

I would like to thank Mike Wooldridge and Wiebe van der Hoek who inspired me to have a closer look at logics for multi-agent systems. To Wiebe van der Hoek (again) for remarks, suggestions and becoming my co-promotor. To Valentin Goranko, for our fruitful cooperation without having ever seen each other (well, *I* have seen him once, but he did not see me, so it does not count). To Rafał Somla, Marek Bednarczyk, Mike Wooldridge (again), Sieuwert van Otterloo, Geert Jonker and Pierre Yves Schobbens, for all the valuable remarks and discussions. To anonymous reviewers from various conferences and journals... To my friends, once more, for sharing the excitement of games with incomplete information – with a deck of cards and a pint of beer preferably; and especially to Mariusz Kowara for his scientific help in the bridge department.

Whom else should I thank? I feel quite certain that I must have forgotten to mention someone important. If You are the one, please accept my apologies, and a grateful THANK YOU.

Contents

Chapter 1

Introduction

Every book tells a story – even if the story sometimes meanders, and the book is full of mathematical symbols. So, let the story begin.

1.1 Agents in Action

Recent years saw a new discipline emerging within the broad field of Artificial Intelligence. *Multi-agent systems* (Weiss, 1999; Wooldridge, 2002) are a philosophical metaphor that induces a specific way of seeing the world, and makes us use agent-oriented vocabulary when describing the phenomena we are interested in – rather than offering a ready-to-use collection of tools and implementation guidelines. Thus, while some researchers present multi-agent systems as a new paradigm for computation or design, we believe that primarily multi-agent systems form a new paradigm for *thinking* and *talking* about the world, and assigning it a specific conceptual structure. They offer a bunch of intuitions that can be useful when the reality around seems to include multiple autonomous entities. Obviously, such intuitions may be useful when studying computational systems and societies of artificial agents, too. We can see components of such systems as being autonomous, perhaps intelligent, definitely active or even pro-active... having some goals and beliefs... et cetera.

A multi-agent system is an environment, inhabited by multiple agents. What is an *agent* then? Despite numerous attempts to answer this question, we are not quite sure if it is well-formed, since it asks in fact for a precise definition of the term "agent". The metaphor of a multi-agent system seems to build on the intuition that *we* are agents – we, humans – and that other entities we study can be just like us to some extent. The usual properties of agents, like autonomy, pro-activeness etc., seem to be secondary: they are results of an introspection rather than primary assumptions we start with. Thus, there seems to be no conclusive definition of an agent – indeed, can we ever come up with such a definition? It is hard to *define* ourselves.

We are not going to define agents nor multi-agent systems in this thesis. We would rather like to look for a vocabulary and a conceptual structure that approximate our intuitions about agents and their communities in a good way.

The focus of this thesis is on agents' decision-making. This is the theme that links the whole "story" together, through all its unexpected changes of direction, and sideline digressions. Agents *act* in their environments, and somehow they should be able to choose the best actions. Or reasonably good actions at least. Plans, strategies, decisions, choices: these are synonyms that refer to an agent (or a group of agents) executing some action. We would like to exploit the insight they can provide. How can they be represented? In what way can they depend on the agent's current view of the world? Because, in order to consider some plan best (or reasonably good), the agent must have some (implicit or explicit) representation of his environment of action.

The title of the thesis has a double meaning. There are many different models that we can use to represent the same reality, and some of them are presented and studied in the first part of the thesis. Moreover, having multiple competing models at hand, agents may be better off combining them in some way, instead of sticking to one of the models and disregarding the others – and this is what the second part of the thesis proposes.

1.2 Models for Agents in Multi-Agent Environments

An agent must have a model of reality in order to make his decisions. The same environments and situations can be modeled using many different methodologies and conceptual apparata. In particular, the models can be *adaptive* – changing their contents, structure, or the way they influence the agent's choices over time – or *normative* – based on some fixed assumptions about the nature of the reality, and the right ways to proceed.[1] The first kind of models is usually obtained through some sort of machine learning, statistical analysis etc.; if the agent can build up accurate knowledge about the environment, he can certainly benefit from adapting his actions to it. Normative models usually assume the worst possible response from the rest of the world. In consequence, they refer to the lower bound of the agent's abilities, and provide the agent with means to play *safe* rather than brilliant.

Each kind of models proposes a set of notions that can be used to explore the reality and reason about it. This thesis is concerned with logic-based (normative) models of multi-agent systems, and the way these models can be combined with adaptive solutions, so the agents can be more flexible in their actions, while still being relatively secure. In consequence, the thesis includes two main tracks. The first track is focused on qualitative models of multi-agent systems, that draw inspiration from modal logics of processes as well as classical game theory. The second track deals with a concept of multi-level modeling of reality, where various models of the same environment can be combined to improve decision making.

[1] We use the term "normative" in the sense that the models we talk about here (e.g. game theory models) impose some *a priori* assumptions about the behavior of the environment in an authoritative way, and prescribe fixed rules of "best behavior" for the agent. We do not mean, however, that the rules and assumptions refer in any way to moral norms, social norms or any other deontic concept.

1.2.1 Modal Logics and Multi-Agent Systems

Logic-based approaches to Artificial Intelligence seem to be presently undervalued by most AI practitioners. This owes much to the fact that logic was believed to deliver the ultimate solution for all basic AI problems for a long time, and the disappointment which came after that. Indeed, it is hard to claim now that we can use logic-based tools (rule-based systems, for instance) to obtain agents that behave in a satisfying way. Despite recent development of logic-based tools for multi-agent systems, their applications restrict mainly to artificial "toy worlds", as opposed to the real world which is usually fuzzy, noisy and, most of all, hard to characterize with a simple mathematical model. However, we believe that mathematical logic – while probably not the best tool for engineering – should still be important in AI research for at least two reasons.

First, it provides us with a vocabulary for *talking* about systems, and gives the vocabulary precise meaning via models and semantic rules. More importantly, mathematical models provide a conceptual apparatus for *thinking* about systems, that can be as well used outside mathematical logic. The second reason is that creating a formal model of a problem makes one realize many (otherwise implicit) assumptions underlying his or her approach to this problem. The assumptions are often given a simplistic treatment in the model (otherwise the models get too complex to be dealt with), yet their implications are usually worth investigating even in this form. Moreover, having made them explicit, one can strive to relax some of them and still use a part of the formal and conceptual machinery – instead of designing solutions completely ad hoc.

Part I of the thesis investigates such oversimplistic, hard to use, and yet highly interesting models of multi-agent systems. We focus on modal logics with their clear and intuitively appealing conceptual machinery of *possible world semantics* (aka *Kripke semantics*). The logics we investigate draw from the long tradition of philosophical studies on human behavior and the behavior of the world in general, that yielded epistemic logic, deontic logic, temporal logic etc. In particular, we investigate Alternating-time Temporal Logic and its extensions, with their conceptual apparatus originating from the classical game theory. As game theory emerged in an attempt to give precise meaning to common-sense notions like choices, strategies, rationality – and to provide formal models of interaction between autonomous entities, it seems a perfect starting point for modeling and reasoning about multi-agent systems.

It should be pointed out that the modal logics for multi-agent systems (and their models) can be used in at least two ways. First, we may strive to represent an objective observer's view to a multi-agent system with the instruments they provide. This is the viewpoint we usually adopt while talking about "specification", "design", "verification" etc. The observer (e.g., the designer or the administrator of the system) may collect all relevant aspects of the system in a Kripke model, and then derive or verify certain properties of this model. Or, the designer can specify some desirable properties of a system, and then try to engineer a model in which those properties hold.

On the other hand, the models can be also used to express a *subjective* view of an agent to the reality he is acting in. In such a case, the agent can represent his knowledge about the world with a model, and ask about properties of the world via the properties of the model, or, more importantly, look for a strategy that makes some desirable property true in the model.

1.2.2 Relaxing the Assumptions to Obtain Adaptivity

Logic-based models (or, more generally, mathematical models) are widely used, and their importance goes well beyond mathematical theories of computing. Decision trees, control flow charts, data flow charts, Q-models, Bayesian nets and classifiers, statistical models, fuzzy sets and fuzzy measures (including possibility and probability measures), and even neural networks to some extent – they all belong to this class. It is probably the inflexibility of decision-making procedures, provided by "pure" mathematical methods, that seems to yield severe limitations for the applicability of the methods to real-life problems. The machine learning approach emphasizes the importance of flexibility and robustness of agents, through an attempt to obtain an accurate and up-to-date model of the world. Models are *adaptive* not because of their inherent structure, but because of the way they are built up and maintained.

An agent can learn to exploit weaknesses of his adversary, to converge with a dynamic, possibly indifferent environment, or to learn trust and cooperation with other agents. In most cases the representation of the environment is quantitative, not qualitative – hence the goal of the agent is to maximize his numerical reward (payoff, utility) in the long run. The popular decision making criterion of expected payoff maximization (with respect to the agent's current knowledge about the environment) shows yet another influence of the mathematical methods from game theory and decision theory. However, no learning algorithm can guarantee an accurate model of the environment, and that is why game theory solutions are still attractive when a wrong decision can bring disastrous results. So, we try to relax the game theory assumptions in Part II of the thesis, in a way that does not give up the security offered by game theory-inspired solutions completely.

1.3 Structure of the Thesis

The thesis is divided into two parts. Part I presents several modal logics that can be used to model and describe agents and their communities. We show the similarities between various languages that have been already proposed, and study how the conceptual machinery they provide matches our intuitive understanding of the notion of agency. As agents are, most of all, supposed to act, we use the "planning as model checking" paradigm to obtain a planning algorithm that can be used within these frameworks.

Part II presents an idea of hierarchical modeling of the reality, and multilevel decision making. In the presentation, we focus on the way the idea can be used to combine the adaptivity of machine learning approaches with the security offered by normative solutions similar to the ones presented in Part I.

1.3.1 Part I: Around Alternating-time Temporal Logic

We use Chapters 2 and 3 to draw parallels between several logics that have been recently proposed to reason about agents and their abilities. These are: coalition game logics CL and ECL introduced by Pauly in 2000, alternating-time temporal logic ATL developed by Alur, Henzinger and Kupferman between 1997 and 2002, and alternating-time temporal epistemic logic ATEL by van der Hoek and Wooldridge (2002), as well as the modal logic of beliefs, desires and intentions (BDI), proposed by Rao and Georgeff in mid-90s. The focus in this part of the thesis is on models: alternating transition systems, multi-player game models (alias concurrent game structures) and coalition effectivity models turn out to be intimately related, while alternating epistemic transition systems and BDI models share much of their philosophical and formal apparatus. Our approach is constructive: we present ways to transform between different types of models and languages.

First, Alternating-time Temporal Logic and Coalition Logic are introduced and discussed in Chapter 2. We present the syntax and semantics of these logics, and show that both their languages and models have very much in common. In the conceptual sense, both CL and ATL build upon branching-time temporal logics like CTL; they both incorporate the game-theoretic notion of strategy, and both ask about what properties can be infallibly *enforced* by which agents or teams. In the formal sense, important subclasses of ATL and CL models can be proved isomorphic, and we can prove that the expressive power of CL is covered by ATL. The chapter is concluded with a simple adaptation of the ATL model checking algorithm so that it can be used for decision making in environments inhabited by multiple agents.

Then, in Chapter 3, Alternating-time Temporal Epistemic Logic is discussed. This logic enriches ATL with an epistemic component to enable modeling (and reasoning about) agents' beliefs under uncertainty. We present a few formal results, relating the conceptual and formal apparatus of ATEL to to these of ATL and the BDI framework, and allowing to use the planning algorithm from Chapter 2 for ATEL and BDI agents as well. Unfortunately, ATEL semantics turns out to be counterintuitive in some respects. In Chapter 4 we show that the notion of allowable strategy under uncertainty should be defined with some caution, and we point out the difference between an agent knowing that he has a suitable strategy and knowing the strategy itself. We also suggest that the agents should be assumed to have similar epistemic capabilities in the semantics of both strategic and epistemic operators. Trying to implement these ideas, we propose two different modifications of ATEL. The first one, dubbed Alternating-time Temporal Observational Logic (ATOL), is a logic for agents with bounded recall of the past. The second, ATEL-R*, is a framework to reason about both perfect and imperfect recall.

The generic framework of ATL can be extended along various dimensions. Another extension of ATL – this time with the notion of agents' obligations – is proposed and discussed in Chapter 5. The way both frameworks are combined is straightforward: we add deontic accessibility relations to ATL models, and deontic operators to the language of ATL (an additional operator $\mathcal{UP}$ is proposed for "unconditionally permitted" properties, similar to the "all I know" operator from epistemic logic). Some formal results are presented; however, we rather focus on demonstrating how obligations of agents can be confronted with their abilities.

1.3.2 Part II: Security vs. Adaptivity, Multilevel Decision Making

Chapter 6 opens the less logically-oriented part of the thesis. It is suggested that an agent does not have to stick to a single model of the reality; instead he can possess a set of complementary beliefs, both learned and assumed, and use them proportionally to the confidence he has in them. A hierarchy of beliefs for an agent is proposed here, together with a decision making scheme. Chapter 7 reports the research on a confidence measure that suits the decision making based on hierarchies of models. We conjecture that there are roughly two sources of doubt that should decrease the agent's confidence in his own beliefs about the world. First, the agent may have too little data. This shows the need for a confidence measure in an obvious way: when a software agent starts interaction with a completely new user, for instance, his knowledge about the user is virtually none – yet it is utilized in the same way by most algorithms regardless of the number of learning steps that have been taken so far. Next, the environment might have changed considerably, so the data do not reflect its current shape. The agent can certainly benefit from detecting conspicuous changes of pattern in the user's behavior, and acting more cautiously in such situations. In order to capture these phenomena, confidence measures based on aggregate variance of the estimator provided by the learning process, and on the self-information loss function are proposed and investigated, with various degree of success.

Chapter 8 presents some experimental results to support the idea. The experiments consisted of the agent's interactions with simulated 0-, 1- and 2-level agents, acting as customers of an imaginary Internet banking service. This is also where both tracks of the thesis come to a joint epilogue: ATL models and planning can be used within the hierarchy of models to induce safer play in a more sophisticated environment. Finally, some concluding remarks are proposed in Chapter 9.

1.3.3 Publications

The thesis builds on a number of papers, and the material from these papers was partially used to form the contents of the thesis. The papers, and the chapters they were used in, are indicated below:

- Chapter 2 uses a part of (Goranko and Jamroga, 2004), and most of (Jamroga, 2004); also, some remarks from (Jamroga, 2003d) and (Jamroga and van der Hoek, 2003) are elaborated there;
- Chapter 3 builds upon another part of (Goranko and Jamroga, 2004).
- Chapter 4 uses most of (Jamroga and van der Hoek, 2004);
- Chapter 5 is based on (Jamroga et al., 2004);
- Chapter 6 builds upon (Jamroga, 2002b) and (Jamroga, 2001b);
- Chapter 7 presents the research already reported in (Jamroga, 2003a), (Jamroga, 2002a) and (Jamroga, 2003b);
- Chapter 8 uses the results from (Jamroga, 2003c) to some extent.

Part I

Logic-Based Modeling of Multi-Agent Systems

Chapter 2

Models and Logics of Strategic Ability

SYNOPSIS. *As stated in Chapter 1, we are going to study agents and their environments within this thesis. The agents' choices, abilities, beliefs, obligations. Ways of modeling the reality around, planning, decision making. Using a popular term: we are going to study agents that act in* multi-agent systems*.*

But – what is a multi-agent system? This question can be answered either in a formal, or an informal way. We investigate several formal *models of multi-agent systems in this chapter, hoping that this can induce some informal understanding of the phenomenon as well.*

2.1 Introduction

In this chapter we offer a comparative analysis of several recent logical enterprises that aim at modeling multi-agent systems. Most of all, the *coalition game logic* CL and its extended version ECL (Pauly, 2002, 2001b,a), and the *Alternating-time Temporal Logic* ATL (Alur et al., 1997, 1998a, 2002) are studied. These turn out to be intimately related, which is not surprising since all of them deal with essentially the same type of scenarios, viz. a *set of agents* (players, system components) taking actions, simultaneously or in turns, on a common set of states – and thus effecting transitions between these states. The game-theoretic aspect is very prominent in both approaches; furthermore, in both frameworks the agents pursue certain goals with their actions and in that pursuit they can form *coalitions*. In both enterprises the objective is to develop formal tools for reasoning about such coalitions of agents and their ability to achieve specified outcomes in these action games.

The study of Alternating-time Temporal Logic and coalition logic, presented in this chapter, forms a basis for the first part of the thesis. The logics have clear possible worlds semantics, are axiomatizable, and have some interesting computational properties. Even these features alone may make them attractive for a logician. However, our motivation goes much beyond that. The logics are underpinned by a clear and intuitively appealing conceptual machinery for talking and *thinking* about systems that involve multiple autonomous agents. The basic notions, used here, originate from classical game theory, which emerged in an attempt to give precise meaning to commonsense notions like choices, strategies, or rationality – and to provide formal models of interaction between autonomous entities, that could be used in further study. Thus, the notions and models were meant to describe real-life phenomena that occur in communities of individual and collective agents (e.g., companies). In fact, game theory have always been considered as much a part of mathematics as it is a part of economics (recall the title of the book by von Neumann and Morgenstern, that gave birth to the whole discipline: "Games in *Economic* Behavior").

Of course, the treatment of interaction, given by von Neumann, Morgenstern and Nash, *is* oversimplistic, and its fundamental philosophical merit has also been questioned.[1] One may even argue whether modeling of intelligent agents and their interaction can be done with the tools of mathematics and formal logic at all (Winograd and Flores, 1986; Pfeifer and Scheier, 1999). However, having a formal model of a problem makes one realize many (otherwise implicit) assumptions underlying his or her approach to this problem. Then – we can study implications of the assumptions, and accept them or revise them (the way we do in Chapter 4); we can extend the models with additional notions (like the notions of knowledge and obligation in Chapters 3 and 5), or we can strive to relax some of the assumptions in a systematic way (cf. Chapters 6, 7 and 8, where a combination of game theory-based and adaptive decision making is studied). Modal logics that embody basic game theory notions – and at the same time build upon (and extend) branching-time temporal logics, well known and studied in the context of computational systems – seem a good starting point for this.

The chapter is organized as follows: first, a brief summary of the basic concepts from game theory and computation tree logic is offered; then we introduce the main "actors" of this study – logics and structures that have been recently proposed for modeling multi-agent systems in a temporal perspective, including all relevant definitions from (Pauly, 2002, 2001a; Alur et al., 1998a, 2002).[2] In Section 2.7, the relationships between these logics and structures are investigated in a formal way. The main results are the following:

- we show that specific classes of multi-player game models (MGMs in short) are equivalent to some types of alternating transition systems (ATSs);
- we show that ATL subsumes CL as well as ECL;
- we show that the three alternative semantics for Alternating-time Temporal Logic and Coalition Logics (based on multi-player game models, alternating transition systems and coalition effectivity models) are equivalent.

Obviously, each of the three alternative semantics for ECL and ATL, investigated here, has its own drawbacks and offers different advantages for practical use. A few remarks on this issue can be found in Section 2.7.4.

[1] Consider this quote from (Shubik, 1998): "Rational Behavior [is]: greed, modified by sloth, constrained by formless fear and justified *ex post* by rationalization."

[2] We make small notational changes here and there to make the differences and common features between the models and languages clearer and easier to see.

The models (and languages) of ATL and CL can be used in at least two ways. First, they may represent an objective observer's view to a multi-agent system. This is the viewpoint we usually adopt while talking about "specification", "design", "verification" etc. The observer (e.g., the designer or the administrator of the system) may collect all relevant aspects of the system in an MGM, and then derive or verify certain properties on this MGM (via model checking, for instance). Or, the designer can specify some desirable properties of a system, and then try to engineer an MGM in which those properties hold (this procedure corresponds to the satisfiability problem).

On the other hand, the models can be also used to express a *subjective* view of an agent to the reality he is acting in. In such a case, the agent can represent his knowledge about the world with an MGM or ATS, and ask about properties of the world via model checking of respective ATL formulae. In particular, agents who use multi-player game models or alternating transition systems can benefit from the "planning as model checking" idea. We show how the ATL model checking algorithm from (Alur et al., 2002) can be adapted to support planning in Section 2.8.

The results from this chapter prepare the ground for subsequent chapters. Since ATL has strictly more expressive power than CL and ECL, we can stick to ATL as our device for reasoning about agents without any loss of generality. As the three alternative semantics of ATL turn out to be equivalent, we can use them (and the semantic structures behind them) interchangeably without paying much attention to the actual choice. This proves very convenient while defining extended languages like ATEL, ATOL or DATL, as well as semantic structures capable to represent agents' beliefs, obligations, and strategic abilities under uncertainty. The subsequent chapters propose also how model checking for ATEL and DATL can be reduced to ATL model checking, yielding an efficient planning procedure for epistemic and deontic goals, too.

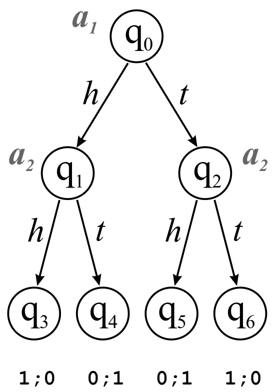
This chapter builds upon a number of papers: most notably (Goranko and Jamroga, 2004), a paper co-written with Valentin Goranko from the Rand Afrikaans University, and (Jamroga, 2004). Also, (Jamroga, 2003d) and (Jamroga and van der Hoek, 2003) were used here to some extent. It should be pointed out that the main equivalence/subsumption results from Sections 2.5 and 2.7 were already published by Goranko in (Goranko, 2001). A similar proof of the equivalence between the ATL semantics based on alternating transition systems and concurrent game structures was proposed independently in (Jamroga and van der Hoek, 2003).

2.2 Basic Influences: Logic Meets Game Theory

ATL and CL have clearly been inspired by some fundamental concepts – coming from both game theory and modal logics of computation – that enable to model and reason about situations in which no uncertainty is taken into account. We try to sketch the concepts in this section.

2.2.1 Classical Game Theory

Logics of agents and action build upon several important concepts from game theory, most of them going back to the 40s and the seminal book (von Neumann and



	q_1hq_2h	q_1hq_2t	q_1tq_2h	q_1tq_2t
q_0h	1;0	1;0	0;1	0;1
q_0t	0;1	1;0	0;1	1;0

Figure 2.1: Extensive and strategic form of the matching pennies game: the perfect information case.

Morgenstern, 1944). We start with an informal survey of these concepts, following mostly Hart (1992). An interested reader is referred to (Aumann and Hart, 1992; Osborne and Rubinstein, 1994) for a more extensive introduction to game theory.

In game theory, a game is usually presented in its extensive and/or strategic form. The *extensive form* defines the game via a tree of possible positions in the game (*states*), game moves (*choices*) available to players, and the outcome (*utility* or *payoff*) that players gain at each of the final states. These games are usually *turn-based*, i.e. every state is assigned a player who controls the choice of the next move, so the players are taking turns. A *strategy* for player a specifies a's choices at the states controlled by a.

The *strategic form* consists of a matrix that presents the payoffs for all combinations of players' strategies. It presents the whole game in a "snapshot" as if it was played in one single move, while the extensive form emphasizes control and information flow in the game.

Example 2.1 Consider a variant of the *matching pennies* game. There are two players, each with a coin: first a_1 chooses to show the heads (action *h*) or tails (*t*), then a_2 does. If both coins are heads up or both coins are tails up, then a_1 wins (and gets score of 1) and a_2 loses (score 0). If the coins show different sides, then a_2 is the winner.

The extensive and strategic forms for this game are shown in Figure 2.1. The strategies define agent's choices at all "his" nodes, and are labeled appropriately: q_1tq_2h denotes, for instance, a strategy for a_2 in which the player chooses to show heads whenever the current state of the game is q_1, and tails at q_2. Note that – using this strategy – a_2 wins regardless of the first move from a_1. □

Section 3.2 shows how the concepts of strategic and extensive game forms can be extended to tackle games that involve players' uncertainty as well.

A general remark is in order here. The concept of coalitional game, traditionally considered in game theory, where every possible coalition is assigned a real number (its *worth*), differs somewhat from the one considered here. In this study we are rather

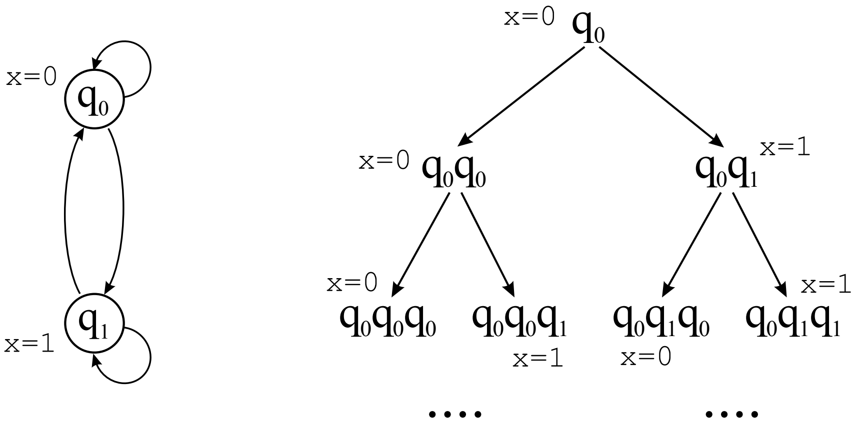


Figure 2.2: Transitions of the variable controller/client system, together with the tree of possible computations.

concerned with *qualitative* aspects of *game structures* rather than with *quantitative* analysis of specific *games*.

2.2.2 Computational Tree Logic

Apart from game theory, the concepts investigated in this chapter are strongly influenced by modal logics of computations, such as the *computation tree logic* CTL. CTL (Emerson, 1990; Huth and Ryan, 2000) involves several operators for temporal properties of computations in transition systems: A (*for all paths*), E (*there is a path*), $\bigcirc$ (*nexttime*), $\Diamond$ (*sometime*), $\Box$ (*always*) and $\mathcal{U}$ (*until*). "Paths" refer to alternative courses of events that may happen in the future; nodes on a path denote states of the system in subsequent moments of time along this particular course. Typically, paths are interpreted as sequences of successive states of computations.

Example 2.2 As an illustration, consider a system with a binary variable x. In every step, the variable can retain or change its value. The states and possible transitions are shown in Figure 2.2. There are two propositions available to observe the value of x: "x=0" and "x=1" (note: these are just atomic propositions, = is not the equality symbol here). Then, for example, E$\Diamond$x=1 is satisfied in every state of the system: there is a path such that x will have the value of 1 at some moment. However, the above is not true for *every* possible course of action: $\neg$A$\Diamond$x=1. $\Box$

It is important to distinguish between the *computational structure*, defined explicitly in the model, and the *behavioral structure*, i.e. the model of how the system is supposed to behave in time (Schnoebelen, 2003). In many temporal models the computational structure is finite, while the implied behavioral structure is infinite. The computational structure can be seen as a way of defining the tree of possible (infinite) computations that may occur in the system. The way the computational structure unravels into a behavioral structure (computation tree) is shown in Figure 2.2, too.

2.2.3 Other Logics of Action and Agency

The logics studied here have a few things in common. They are intended for reasoning about various aspects of multi-agent systems and multi-player games, they are multi-modal logics, they have been obviously inspired by game theory, and they are based on the temporal logic approach. A number of other proposals, such as the dynamic logic-based *intention logic* (Cohen and Levesque, 1990), the KARO framework (van Linder et al., 1998) or the *dynamic epistemic logic* (van Benthem, 2001), will not be discussed here. A broader survey of logic-based approaches to multi-agent systems can be found in (Fagin et al., 1995) and (van der Hoek and Wooldridge, 2003c).

One related body of work, however, should be briefly mentioned: namely, the "stit" logic – the logic of *seeing to it that* (Belnap and Perloff, 1988; Belnap, 1991). Such logics contain an *agentive* modality, which attempts to capture the idea of an agent *causing* some state of affairs. This modality, typically written $[i \text{ stit } \varphi]$, is read as "agent i sees to it that φ". The semantics of stit modalities are typically given as $[i \text{ stit } \varphi]$ iff i makes a choice c, and φ is a necessary consequence of choice c (i.e., φ holds in all futures that could arise through i making choice c). A distinction is sometimes made between the "generic" stit modality and the *deliberate* stit modality "dstit" (Horty and Belnap, 1995); the idea is that i deliberately sees to it that φ if $[i \text{ stit } \varphi]$ and there is at least one future in which φ does not hold (the intuition being that i is then making a *deliberate choice* for φ, as φ would not necessarily hold if i did not make choice c). The logics of ATL and CL, which we study in the following sections, embody somewhat similar concerns. However, they are underlain by fundamentally different semantic constructs. Moreover, stit formulae assert that an agent *makes* a particular choice, whereas we have no direct way of expressing this in ATL nor CL.

2.3 Coalition Logics and Multi-Player Game Models

Coalition logic (CL), introduced in (Pauly, 2001b, 2002), formalizes reasoning about powers of coalitions in strategic games. It extends classical propositional logic with a family of (non-normal) modalities $[A]$, $A \subseteq \mathbb{A}\text{gt}$, where $\mathbb{A}\text{gt}$ is a fixed set of players. Intuitively, $[A]\varphi$ means that coalition A can *enforce* an outcome state satisfying φ.

2.3.1 Multi-Player Strategic Game Models

Game frames (Pauly, 2002), represent multi-player strategic games where sets of players can form coalitions in attempts to achieve desirable outcomes. Game frames are based on the notion of a *strategic game form* – a tuple $\langle \mathbb{A}\text{gt}, \{\Sigma_a \mid a \in \mathbb{A}\text{gt}\}, Q, o\rangle$ consisting of:

- a non-empty finite set of *agents* (or *players*) $\mathbb{A}\text{gt}$,

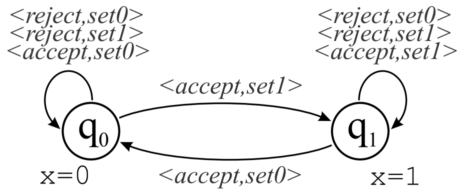


Figure 2.3: Transitions of the variable controller/client system.

- a family of (non-empty) sets of *actions* (*choices*, *strategies*) Σ_a for each player $a \in \mathbb{A}\mathrm{gt}$,
- a non-empty set of *states* Q,
- an *outcome function* $o : \prod_{a \in \mathbb{A}\mathrm{gt}} \Sigma_a \to Q$ which associates an outcome state in Q to every combination of choices from all the players. By a *collective choice* σ_A, we denote a tuple of choices $\langle \sigma_a \rangle_{a \in A}$ (one for each player from $A \subseteq \mathbb{A}\mathrm{gt}$), and we write $o(\sigma_A, \sigma_{\mathbb{A}\mathrm{gt} \setminus A})$ with the presumed meaning.

Remark 2.1 *Elements of set Σ_a were originally called* strategies *in (Pauly, 2001b, 2002). Note that this notion of a "strategy" is local, wrapped into one-step actions. It differs from the notion of a "strategy" in an extensive game form (used in the semantics of* ATL*) which represents a global, conditional plan of action. To avoid confusion, we refer to the local strategies as* actions *or* choices*, and use the term* collective choice *instead of* strategy profile *from (Pauly, 2002) to denote a combination of simultaneous choices from several players.*

Remark 2.2 *A strategic game form defines the choices and transitions available at a particular state of the game. If the identity of the state does not follow from the context in an obvious way, we use indices to indicate which state they refer to.*

The set of all strategic game forms for players $\mathbb{A}\mathrm{gt}$ over states Q is denoted by $\Gamma_Q^{\mathbb{A}\mathrm{gt}}$. A *multi-player game frame* for a set of players $\mathbb{A}\mathrm{gt}$ is a pair $\langle Q, \gamma \rangle$ where $\gamma : Q \to \Gamma_Q^{\mathbb{A}\mathrm{gt}}$ is a mapping associating a strategic game form with each state in Q. A *multi-player game model* (MGM) for a set of players $\mathbb{A}\mathrm{gt}$ over a set of propositions Π is a triple $M = \langle Q, \gamma, \pi \rangle$ where $\langle Q, \gamma \rangle$ is a multi-player game frame, and $\pi : Q \to \mathcal{P}(\Pi)$ is a *valuation* labeling each state from Q with the set of propositions that are true at that state.

Example 2.3 Consider a variation of the system with binary variable x from Example 2.2. There are two processes: the controller (or server) s can enforce the variable to retain its value in the next step, or let the client change the value. The client c can request the value of x to be 0 or 1. The players proceed with their choices simultaneously. The multi-player game model for this system consists of two game forms, defining choices and transitions for states q_0 and q_1 respectively; the states and transitions of the system as a whole are shown in Figure 2.3. Again, we should make the distinction between computational and behavioral structures. The multi-player game

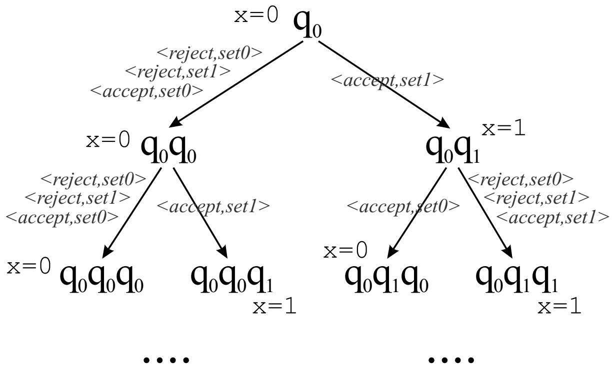


Figure 2.4: The tree of possible computations (assuming that q_0 is the initial state).

model unravels into a computation tree in a way analogous to CTL models (Figure 2.4).
□

2.3.2 Coalition Logic

Formulae of CL are defined recursively as:

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \psi \mid [A]\varphi.$$

where $p \in \Pi$ is a proposition, and $A \subseteq \mathbb{A}\mathrm{gt}$ is a group of agents. Every proposition can be true in some states of the system, and false in the others; the exact truth values for a particular multi-player game model M are given by function π. Coalitional modalities $[A]$ are the novelty here: the informal meaning behind $[A]\varphi$ is that agents A can cooperate to ensure that the outcome of the (one-step) game satisfies φ.

Formally, the semantics of CL can be given via the clauses:

$M, q \models p$	iff	$p \in \pi(q)$ for atomic propositions p;
$M, q \models \neg\varphi$	iff	$M, q \not\models \varphi$;
$M, q \models \varphi \vee \psi$	iff	$M, q \models \varphi$ or $M, q \models \psi$;
$M, q \models [A]\varphi$	iff	there is a collective choice σ_A such that for every collective choice $\sigma_{\mathbb{A}\mathrm{gt}\setminus A}$, we have $M, o_q(\sigma_A, \sigma_{\mathbb{A}\mathrm{gt}\setminus A}) \models \varphi$.

Example 2.4 Consider the variable client/server system from Example 2.3. The following CL formulae are valid in this model (i.e. true in every state of it):

1. $(\mathsf{x{=}0} \rightarrow [s]\mathsf{x{=}0}) \wedge (\mathsf{x{=}1} \rightarrow [s]\mathsf{x{=}1})$: the server can enforce the value of x to remain the same in the next step;

2. $\mathsf{x{=}0} \rightarrow \neg[c]\mathsf{x{=}1}$: c cannot change the value from 0 to 1 on his own;

3. $\mathsf{x{=}0} \rightarrow \neg[s]\mathsf{x{=}1}$: s cannot change the value on his own either;

4. x=0 $\rightarrow [s, c]$x=1 : s and c can cooperate to change the value.

□

2.3.3 Logics for Local and Global Effectivity of Coalitions

In CL, operators $[A]$ can be seen as expressing *local effectivity* of coalitions, i.e. their powers to force outcomes in a single game. In this sense, CL can be thought of as reasoning about *strategic game forms*. Pauly extends CL to the *Extended Coalition Logic* ECL in (Pauly, 2001b), with operators for iterative *global effectivity* of agents. $[A^*]\varphi$ says that coalition A has a collective strategy to maintain the truth of φ in a collection of games *played repeatedly ad infinitum*. Alternatively, we can see operators $[A]$ as a formalization of reasoning about a single move in a (possibly more complex) game, and $[A^*]$ as referring to an analysis of the entire game. In this case, both CL and ECL formalize reasoning about different aspects of *extensive game forms*, representing sequences of moves, collectively effected by the players' actions.

Since ECL can be embedded as a fragment of ATL (as presented in Section 2.5), we will not discuss it separately here.

2.4 Alternating-Time Temporal Logic and Its Models

Game-theoretic scenarios can occur in various situations, one of them being *open computer systems* such as computer networks, where the different components can act as relatively autonomous agents, and computations in such systems are effected by their combined actions. The *Alternating-time Temporal Logics* ATL and ATL*, introduced in (Alur et al., 1997), and later refined in (Alur et al., 1998a, 2002), are intended to formalize reasoning about computations in such open systems which can be enforced by coalitions of agents, in a way generalizing the logics CTL and CTL*.

2.4.1 The Full Logic of ATL*

In ATL* a class of *cooperation modalities* $\langle\langle A\rangle\rangle$ replaces the path quantifiers E and A. The common-sense reading of $\langle\langle A\rangle\rangle\Phi$ is:

> "The group of agents A have a collective strategy to enforce Φ regardless of what all the other agents do".

Φ can be any temporal formula that refers to properties of a path (so called *path formula*). A dual operator $[\![A]\!]$ can be defined in the usual way as $[\![A]\!]\Phi \equiv \neg\langle\langle A\rangle\rangle\neg\Phi$, meaning that A cannot avoid Φ on their own. The original CTL* operators E and A can be expressed in ATL* with $\langle\langle \mathbb{A}\mathrm{gt}\rangle\rangle$ and $\langle\langle\varnothing\rangle\rangle$ respectively, but between both extremes one can express much more about the abilities of particular agents and groups of agents. ATL* inherits all the temporal operators from CTL*: $\bigcirc$ (*nexttime*), $\Diamond$ (*sometime*), $\Box$ (*always*) and $\mathcal{U}$ (*until*).

The full, unrestricted version of Alternating-time Temporal Logic ATL* consists of state formulae and path formulae. A state formula is one of the following:

- p, where p is an atomic proposition;
- $\neg\varphi$ or $\varphi \vee \psi$, where φ, ψ are ATL* state formulae;
- $\langle\langle A\rangle\rangle\Phi$, where $A \subseteq \mathbb{A}\mathrm{gt}$ is a set of agents, and Φ is an ATL* path formula.

A path formula is one of the following:

- an ATL* state formula;
- $\neg\varphi$ or $\varphi \vee \psi$, where φ, ψ are ATL* path formulae;
- $\bigcirc\varphi$ or $\varphi\mathcal{U}\psi$, where φ, ψ are ATL* path formulae.

The temporal operators "sometime" ($\Diamond$) and "always" ($\Box$) can be defined as:

$$\begin{aligned} \Diamond\varphi &\equiv \top\,\mathcal{U}\varphi, \text{ and} \\ \Box\varphi &\equiv \neg\Diamond\neg\varphi. \end{aligned}$$

2.4.2 "Vanilla" ATL

In "vanilla" ATL (i.e. ATL without *) it is required that every occurrence of a temporal operator is preceded by exactly one occurrence of a cooperation modality (that is, ATL is the fragment of ATL* subjected to the same syntactic restrictions which define CTL as a fragment of CTL*). In consequence, only state formulae can be found in ATL: p, $\neg\varphi$, $\varphi \vee \psi$, $\langle\langle A\rangle\rangle\bigcirc\varphi$, $\langle\langle A\rangle\rangle\Box\varphi$, and $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$, where p is an atomic proposition, φ, ψ are ATL formulae, and A is a coalition of agents. Since model-checking for ATL* requires 2EXPTIME, but it is linear for ATL, ATL is more useful for practical applications, and we will rather focus on ATL here. Formally, the recursive definition of ATL formulae is:

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

The "sometime" operator $\Diamond$ can be defined in the usual way as:

$$\langle\langle A\rangle\rangle\Diamond\varphi \equiv \langle\langle A\rangle\rangle\top\,\mathcal{U}\varphi.$$

Examples of interesting properties that can be expressed with ATL include:

1. $\langle\langle A\rangle\rangle\Diamond\varphi$
2. $\langle\langle A\rangle\rangle\Box\varphi$
3. $\neg\langle\langle A\rangle\rangle\bigcirc\varphi \wedge \neg\langle\langle B\rangle\rangle\bigcirc\varphi \wedge \langle\langle A \cup B\rangle\rangle\bigcirc\varphi$
4. $\langle\langle A \cup \{a\}\rangle\rangle\bigcirc\varphi \rightarrow \langle\langle\{a\}\rangle\rangle\bigcirc\varphi$

The first of these expresses a kind of *cooperative liveness* property: coalition A can assure that eventually some ATL-formula φ will hold. The second item then expresses a *cooperative safety* property: A can ensure that φ is an invariant of the system. The third item is an example of what coalitions can achieve by forming bigger ones; although coalition A and B both cannot achieve that in the next state φ will be true, if they *joined their forces*, they would have a strategy to enforce φ in the next state. Finally, the last property expresses that a does not need any partner from A to achieve that φ will hold in the next state: read as a scheme, it says that whatever A together with a can achieve next, can be achieved by a on his own.

It should be noted that at least three different versions of semantic structures for ATL have been proposed by Alur and colleagues in the last 7 years. The earliest version (Alur et al., 1997), includes definitions of synchronous turn-based structures and asynchronous structures in which every transition is controlled by a single agent. The next paper (Alur et al., 1998a) defines general structures, called *alternating transition systems*, where agents' choices are identified with the sets of possible outcomes. In *concurrent game structures* from (Alur et al., 2002), labels for choices are introduced and the transition function is simplified; moreover, an arbitrary finite set of agents $\mathbb{A}\mathrm{gt}$ is replaced with set $\{1, ..., k\}$.

The above papers share the same title and they are often cited incorrectly in the literature as well as citation indices, which may lead to some confusion.

Remark 2.3 *The version of* ATL *from (Alur et al., 1997) is somewhat preliminary: there is no concurrency possible in the models, as they are limited to the turn-based case only (every transition is governed by a single agent). The version has distinctly less expressive power than the other two – many examples of games that are not turn-based and can be modeled with the later versions of* ATL *can be found, for instance, in (Alur et al., 2002). Therefore we will discuss only the later versions (Alur et al., 1998a, 2002) through the rest of the chapter.*

Remark 2.4 *The complexity results for* ATL *model checking look very attractive at the first glance: given model M, the set of all states q such that $M, q \models \varphi$ can be computed in time $O(ml)$, where m is the number of transitions in M, and l is the length of formula φ (Alur et al., 2002). It should be pointed out, however – and has been pointed out in (Alur et al., 2002), too – that while the problem is linear* in the size of the structure*, the structure itself can be very large. In fact, in the simplest case when we identify the states in the model with the combinations of values of n Boolean variables, the model has 2^n states. In other words, the model can be exponentially large in the number of dimensions of the problem.*

Consider the rocket example from Section 2.8.3, and the model presented in Figure 2.13. There are only three (binary) dimensions to this problem: the rocket can be either in London or in Paris, its tank can be either full or empty, and the cargo can be in or out. A multi-player game model that describes this domain is exponentially large: it has $2^3 = 8$ states and 300 *transitions. Unfortunately, the explosion is unavoidable in the general case, although there is some ongoing research on a more compact representation for* ATL *domains, that does not suffer from the exponential explosion of states in some situations (Kacprzak and Penczek, 2004).*

2.4.3 Alternating Transition Systems

Alternating transition systems – building on the concept of *alternation* developed in (Chandra et al., 1981) – formalize systems of transitions effected by collective actions of all agents involved. In the particular case of one agent (the *system*), alternating transition systems are reduced to ordinary transition systems, and ATL reduces to CTL.

An *alternating transition system* (ATS) is a tuple $T = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, \delta \rangle$ where:

- $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$ is a non-empty, finite set of *agents*,
- Q is a non-empty set of *states*,
- Π is a set of (atomic) *propositions*,
- $\pi : Q \to \mathcal{P}(\Pi)$ is a *valuation* of propositions, and
- $\delta : Q \times \mathbb{A}\mathrm{gt} \to \mathcal{P}(\mathcal{P}(Q))$ is a *transition function* mapping a pair $\langle state, agent \rangle$ to a non-empty family of choices of possible next states. The idea is that $\delta(q, a) = \{Q_1, ..., Q_n\}$ $(Q_1, ..., Q_n \subseteq Q)$ defines the possible outcomes of agent a's decisions at state q, and the decisions are identified with the outcome sets. When a chooses a set $Q_a \in \delta(q, a)$ at state q, he forces the outcome state to be from Q_a. The resulting transition leads to a state which is in the intersection of all Q_a for $a \in \mathbb{A}\mathrm{gt}$ and so it reflects the mutual will of all agents. Since the system is required to be deterministic (given the state and the agents' decisions), $Q_{a_1} \cap ... \cap Q_{a_k}$ must always be a singleton.

Example 2.5 An ATS for the variable client/server system is shown in Figure 2.5. Note that the transition system includes more states and transitions than the multi-player game model from Example 2.3. Now, the states encode the value of $\mathtt{x}$ *and* the last action made: q_0 refers to "x=0 by s's force", q_0' to "x=0 by c's request" etc. In fact, no ATS with only 2 states exists for this problem – we will prove this formally in Section 2.7.4 (see Proposition 2.20). □

Remark 2.5 *It seems worth pointing out that the way agents' choices are represented (and the way they imply system transitions) is somewhat similar to the concept of* refusals *and* ready sets *from (Hoare, 1985). There, ready sets of a process P include events that can be executed by P, and the parallel composition of processes P_1 and P_2 yields ready sets that are intersections of P_1's and P_2's ready sets – although no assumption about determinism is being made.*

Remark 2.6 *Note also that – despite the singleton requirement – determinism is not a crucial issue with alternating transition systems, as it can be easily modeled by introducing a new, fictitious agent (we may call the agent "nature" or "environment"). Then we can attribute our uncertainty about the outcome of collective choices from all the "real" players to the decisions of this additional player.*

Definition 2.1 *A state $q_2 \in Q$ is a* successor *of q_1 if, whenever the system is in q_1, the agents can cooperate so that the next state is q_2, i.e. there are choice sets $Q_a \in \delta(q_1, a)$, for each $a \in \mathbb{A}\mathrm{gt}$ such that $\bigcap_{a \in \mathbb{A}\mathrm{gt}} Q_a = \{q_2\}$. The set of successors of q is denoted by Q_q^{suc}.*

Definition 2.2 *A* computation *in T is an infinite sequence of states $q_0 q_1 ...$ such that q_{i+1} is a successor of q_i for every $i \geq 0$. A q-*computation *is a computation starting from q.*

(A)

$\mathbb{A}\text{gt} = \{s, c\}$
$Q = \{q_0, q'_0, q_1, q'_1\}$

$\Pi = \{\text{x=0}, \text{x=1}\}$
$\pi(q_0) = \pi(q'_0) = \{\text{x=0}\}$
$\pi(q_1) = \pi(q'_1) = \{\text{x=1}\}$

$\delta(q_0, s) = \delta(q'_0, s) = \{\{q_0\}, \{q'_0, q'_1\}\}$
$\delta(q_1, s) = \delta(q'_1, s) = \{\{q_1\}, \{q'_0, q'_1\}\}$
$\delta(q_0, c) = \delta(q'_0, c) = \{\{q_0, q'_0\}, \{q_0, q'_1\}\}$
$\delta(q_1, c) = \delta(q'_1, c) = \{\{q_1, q'_0\}, \{q_1, q'_1\}\}$

(B)

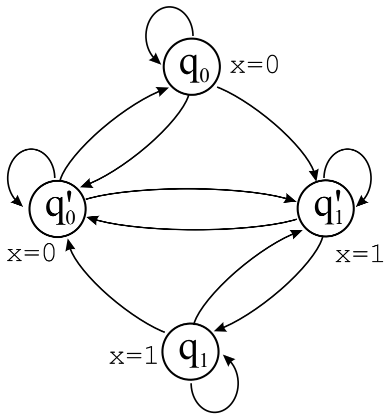


Figure 2.5: An ATS for the controller/client problem: (A) algebraic definition; (B) temporal structure of the system: states, transitions, and valuation of propositions.

2.4.4 Semantics of ATL Based on Alternating Transition Systems

Definition 2.3 *A* strategy *for agent a is a mapping $f_a : Q^+ \to \mathcal{P}(Q)$ which assigns to every non-empty sequence of states $q_0, ..., q_n$ a choice set $f_a(q_0...q_n) \in \delta(q_n, a)$. The function specifies a's decisions for every possible (finite) history of system states. A* collective strategy *for a set of agents $A \subseteq \mathbb{A}\text{gt}$ is just a tuple of strategies (one per agent from A): $F_A = \langle f_a \rangle_{a \in A}$.*

Now, $out(q, F_A)$ denotes the set of all possible (infinite) computations, starting from state q and consistent with F_A, i.e. the set of all q-computations in which group A has been using F_A. More formally, computation $\Lambda = q_0 q_1 ...$ is consistent with a (collective) strategy $F_A = \langle f_a \rangle_{a \in A}$ if, for every $i = 0, 1, ...$, there exists a tuple of agents' decisions $Q^i_{a_j} \in \delta(q_i, a_j)$ for $j = 1, ..., k$, such that $Q^i_{a_1} \cap ... \cap Q^i_{a_k} = q_{i+1}$ and $Q^i_a = f_a(q_0...q_i)$ for each $a \in A$.

Let $\Lambda[i]$ denote the ith position in computation Λ. The definition of truth of an ATL formula at state q of an ATS $T = \langle \Pi, \mathbb{A}\text{gt}, Q, \pi, \delta \rangle$ follows through the clauses given below.

$T, q \models p$ iff $p \in \pi(q)$, for an atomic proposition p;

$T, q \models \neg\varphi$ iff $T, q \not\models \varphi$;

$T, q \models \varphi \vee \psi$ iff $T, q \models \varphi$ or $T, q \models \psi$;

$T, q \models \langle\langle A \rangle\rangle \bigcirc \varphi$	iff	there exists a collective strategy F_A such that for every computation $\Lambda \in out(q, F_A)$ we have $T, \Lambda[1] \models \varphi$;
$T, q \models \langle\langle A \rangle\rangle \Box \varphi$	iff	there exists a collective strategy F_A such that for every $\Lambda \in out(q, F_A)$ we have $T, \Lambda[i] \models \varphi$ for every $i \geq 0$;
$T, q \models \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$	iff	there exists a collective strategy F_A such that for every $\Lambda \in out(q, F_A)$ there is $i \geq 0$ such that $T, \Lambda[i] \models \psi$ and for all j such that $0 \leq j < i$ we have $T, \Lambda[j] \models \varphi$.

Remark 2.7 *This notion of strategy can be specified as "perfect recall strategy", where the whole history of the game is considered when the choice of the next move is made by the agents. The other extreme alternative is a "memoryless strategy" where only the current state is taken in consideration; further variations on "limited memory span strategies" are possible. While the choice of one or another notion of strategy affects the semantics of the full* ATL**, it is not difficult to see that perfect recall strategies and memoryless strategies eventually yield equivalent semantics for* ATL *– cf. also (Schobbens, 2003).*

Remark 2.8 *Note also that a strategy represents what is called a* conditional *or* universal *plan in planning literature, because it does not propose a fixed sequence of actions, but rather describes what the agent should do in every possible situation.*

Example 2.6 The following ATL formulae are valid in the ATS from Figure 2.5:

1. $(\mathsf{x{=}0} \rightarrow \langle\langle s \rangle\rangle \bigcirc\ \mathsf{x{=}0}) \wedge (\mathsf{x{=}1} \rightarrow \langle\langle s \rangle\rangle \bigcirc\ \mathsf{x{=}1})$: the server can enforce the value of x to remain the same in the next step;
2. $\mathsf{x{=}0} \rightarrow (\neg\langle\langle c \rangle\rangle \Diamond\ \mathsf{x{=}1} \wedge \neg\langle\langle s \rangle\rangle \Diamond\ \mathsf{x{=}1})$: neither c nor s can change the value from 0 to 1, even in multiple steps;
3. $\mathsf{x{=}0} \rightarrow \langle\langle s, c \rangle\rangle \Diamond\ \mathsf{x{=}1}$: s and c can cooperate to change the value.

□

2.4.5 Semantics of ATL Based on Concurrent Game Structures and multi-player game models

Alur et al. (2002) redefines ATL models as *concurrent game structures*:

$$M = \langle k, Q, \Pi, \pi, d, o \rangle,$$

where:

- k is a natural number defining the amount of players (so the players are identified with numbers $1, ..., k$ and the set of players $\mathbb{A}\mathrm{gt}$ can be taken to be $\{1, ..., k\}$),
- Q is a finite set of (global) states of the system,
- Π is the set of atomic propositions, and $\pi : Q \rightarrow \mathcal{P}(\Pi)$ is a mapping that specifies which propositions are true in which states.

- The decisions available to player a at state q are labeled with consecutive natural numbers, and function $d : \mathbb{A}\text{gt} \times Q \rightarrow \mathbb{N}$ specifies how many options are available for a particular agent at a particular state. Thus, agent a at state q can choose his decision from set $\{1, ..., d_a(q)\}$. Finally, a complete tuple of decisions $\langle \alpha_1, ..., \alpha_k \rangle$ at state q implies a deterministic transition according to the transition function $o(q, \alpha_1, ..., \alpha_k)$.

In a concurrent game structure, the type of a strategy function slightly differs from the one in an ATS, since choices are abstract entities indexed by natural numbers now, and a strategy is a mapping $f_a : Q^+ \rightarrow \mathbb{N}$ such that $f_a(\lambda q) \leq d_a(q)$. The rest of the semantics looks exactly the same as for alternating transition systems.

Remark 2.9 *Clearly, concurrent game structures are equivalent to Pauly's multi-player game models; they differ from each other only in notation.*[3] *Thus, the* ATL *semantics can be as well based on MGMs, and the truth definitions look exactly the same as for alternating transition systems (see Section 2.4.4). We leave rewriting the definitions of a strategy, collective strategy and outcome set in terms of multi-player game models to the reader. The next section shows how this shared semantics can be used to show that* ATL *subsumes coalition logics.*

2.4.6 Semantics of ATL*

Semantics of the full language ATL* can be defined in a similar way:

$T, q \models \langle\langle A \rangle\rangle \varphi$ iff there exists a collective strategy F_A such that $T, \Lambda \models \varphi$ for all computations $\Lambda \in out(q, F_A)$.

In other words, no matter what the rest of the agents decides to do, the agents from A have a way of enforcing Φ along the resulting course of events. The rest of the semantics is the same as in CTL*. Let Λ^i denote the ith suffix of Λ, i.e. $\Lambda^i = q_i q_{i+1} \ldots$ for $\Lambda = q_0 q_1 \ldots$. Then:

$T, \Lambda \models \varphi$ iff $T, \Lambda[0] \models \varphi$, for φ being a state formula;

$T, \Lambda \models \bigcirc\varphi$ iff $T, \Lambda^1 \models \varphi$;

$T, \Lambda \models \varphi\mathcal{U}\psi$ iff there exists $i \geq 0$ such that $T, \Lambda^i \models \psi$ and for all j such that $0 \leq j < i$ we have $T, \Lambda^j \models \varphi$.

2.5 Embedding CL and ECL into ATL

It turns out that both CL and ECL are strictly subsumed by ATL in terms of the shared semantics based on multi-player game models. Indeed, there is a translation of the formulae of ECL into ATL, which becomes obvious once the ATL semantic clause for $\langle\langle A \rangle\rangle \bigcirc \varphi$ is rephrased as:

[3] The only real difference is that the set of states Q and the sets representing agents' choices are explicitly required to be finite in the concurrent game structures, while MGMs and ATSs are not constrained this way. However, these requirements are not essential and can be easily omitted if necessary.

$T, q \models \langle\langle A\rangle\rangle \bigcirc \varphi$ iff there exists a collective strategy $F_A = \langle f_a \rangle_{a \in A}$ such that for every collective strategy $F_{\mathbb{A}\mathrm{gt} \setminus A} = \langle f_a \rangle_{a \in \mathbb{A}\mathrm{gt} \setminus A}$, we have $T, s \models \varphi$, where $\{s\} = \bigcap_{a \in A} f_a(q) \cap \bigcap_{a \in \mathbb{A}\mathrm{gt} \setminus A} f_a(q)$,

which is equivalent to the truth-condition for $[A]\varphi$ in the coalition logic CL.

Thus, CL embeds in a straightforward way as a simple fragment of ATL by translating $[A]\varphi$ into $\langle\langle A\rangle\rangle \bigcirc \varphi$. Accordingly, $[A^*]\varphi$ translates into ATL as $\langle\langle A\rangle\rangle \Box \varphi$, which follows from the fact that each of $[A^*]\varphi$ and $\langle\langle A\rangle\rangle \Box \varphi$, is the greatest fixpoint of the same operator over $[A]\varphi$ and $\langle\langle A\rangle\rangle \bigcirc \varphi$ respectively (see Section 2.6). In consequence, ATL subsumes ECL as the fragment ATL_{XG} involving only $\langle\langle A\rangle\rangle \bigcirc \varphi$ and $\langle\langle A\rangle\rangle \Box \varphi$.

We will focus on ATL, and will simply regard CL and ECL as its fragments throughout the rest of the thesis.

Remark 2.10 *Note that the coalition logic-related notions of choice and collective choice can be readily expressed in terms of alternating transition systems, which immediately leads to a semantics for* CL *based on ATS, too. Thus,* ATL *and the coalition logics share the semantics based on alternating transition systems as well.*

2.6 Effectivity Functions as Alternative Semantics for ATL

As mentioned earlier, game theory usually measures the powers of coalitions *quantitatively*, and characterizes the possible outcomes in terms of *payoff profiles*. That approach can be easily transformed into a *qualitative* one, where the payoff profiles are encoded in the outcome states themselves and each coalition is assigned a *preference order* on these outcome states. Then, the power of a coalition can be measured in terms of *sets of states* in which it can force the actual outcome of the game (i.e. sets for which it is *effective*), thus defining another semantics for ATL, based on so called *coalition effectivity models* (introduced by Pauly for the coalition logics CL and ECL). This semantics is essentially a monotone neighborhood semantics for non-normal multi-modal logics, and therefore it enables the results, methods and techniques already developed for modal logics to be applied here as well.

Definition 2.4 (Pauly, 2002) *A (local)* effectivity function *is a mapping of type* $e : \mathcal{P}(\mathbb{A}\mathrm{gt}) \to \mathcal{P}(\mathcal{P}(Q))$.

The idea is that we associate with each set of players the family of outcome sets for which their coalition is effective. However, the notion of effectivity function as defined above is abstract and not every effectivity function corresponds to a real strategic game form. Those which do can be characterized with the following conditions (Pauly, 2002):

1. *Liveness:* for every $A \subseteq \mathbb{A}\mathrm{gt}$, $\varnothing \notin e(A)$.
2. *Termination:* for every $A \subseteq \mathbb{A}\mathrm{gt}$, $Q \in e(A)$.

$\varnothing$	$\{s\}$	$\{c\}$	$\{s, c\}$
$\{\{q_0, q_1\}\}$	$\{\{q_0\}, \{q_0, q_1\}\}$	$\{\{q_0\}, \{q_0, q_1\}\}$	$\{\{q_0\}, \{q_1\}, \{q_0, q_1\}\}$

Figure 2.6: A coalition effectivity function for the variable client/server system.

3. $\mathbb{A}\mathrm{gt}$-*maximality:* if $X \notin e(\mathbb{A}\mathrm{gt})$ then $Q \setminus X \in e(\varnothing)$ (if X cannot be effected by the grand coalition of players, then $Q \setminus X$ is inevitable).
4. *Outcome-monotonicity:* if $X \subseteq Y$ and $X \in e(A)$ then $Y \in e(A)$.
5. *Super-additivity:* for all $A_1, A_2 \subseteq \mathbb{A}\mathrm{gt}$ and $X_1, X_2 \subseteq Q$, if $A_1 \cap A_2 = \varnothing$, $X_1 \in e(A_1)$, and $X_2 \in e(A_2)$, then $X_1 \cap X_2 \in e(A_1 \cup A_2)$.

We note that super-additivity and liveness imply *consistency of the powers*: for any $A \subseteq \mathbb{A}\mathrm{gt}$, if $X \in e(A)$ then $Q \setminus X \notin e(\mathbb{A}\mathrm{gt} \setminus A)$.

Definition 2.5 (Pauly, 2002) *An effectivity function e is called* playable *if conditions (1)–(5) hold for e.*

Definition 2.6 (Pauly, 2002) *An effectivity function e is the effectivity function of a strategic game form γ if it associates with each set of players A from γ the family of outcome sets $\{Q_1, Q_2, ...\}$, such that for every Q_i the coalition A has a collective choice to ensure that the next state will be in Q_i.*

Theorem 2.11 (Pauly, 2002) *An effectivity function is playable iff it is the effectivity function of some strategic game form.*

Example 2.7 Figure 2.6 presents a playable effectivity function that describes powers of all the possible coalitions for the variable server/client system from Example 2.3, and state q_0. □

Definition 2.7 (Pauly, 2002) *A* coalition effectivity frame *is a triple $\mathcal{F} = \langle \mathbb{A}\mathrm{gt}, Q, E \rangle$ where $\mathbb{A}\mathrm{gt}$ is a set of players, Q is a non-empty set of states and $E : Q \to (\mathcal{P}(\mathbb{A}\mathrm{gt}) \to \mathcal{P}(\mathcal{P}(Q)))$ is a mapping which associates an effectivity function with each state. We shall write $E_q(A)$ instead of $E(q)(A)$. A* coalition effectivity model *(CEM) is a tuple $\mathcal{E} = \langle \mathbb{A}\mathrm{gt}, Q, E, \pi \rangle$ where $\langle \mathbb{A}\mathrm{gt}, Q, E \rangle$ is a coalition effectivity frame and π is a valuation of the atomic propositions over Q.*

Definition 2.8 *A coalition effectivity frame (resp. coalition effectivity model) is* standard *if it contains only playable effectivity functions.*

Definition 2.9 *A multi-player game model M* implements *a coalition effectivity model $\mathcal{E}$ if $\mathcal{E}$ consists of effectivity functions of the game frames from M.*

Corollary 2.12 *A coalition effectivity model is standard iff it is implemented by some strategic game model.*

Thus, coalition effectivity models provide semantics of CL by means of the following truth definition (Pauly, 2002):

$$\mathcal{E}, q \models [A]\varphi \quad \text{iff} \quad \{s \in \mathcal{E} \mid \mathcal{E}, s \models \varphi\} \in E_q(A).$$

This semantics can be accordingly extended to semantics for ECL (Pauly, 2001a) and ATL (Goranko, 2001) by defining effectivity functions for the global effectivity operators in extensive game forms, where they indicate the outcome sets for which the coalitions have long-term *strategies* to effect. This extension can be done using characterizations of $\langle\langle A\rangle\rangle\Box\varphi$ and $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$ with the greatest fixpoint operator ν and the least fixpoint operator μ respectively. First, let us observe that the following equivalences are valid (i.e. true in every state of every coalition effectivity model):

$$\begin{aligned} \langle\langle A\rangle\rangle\Box\varphi &\leftrightarrow \varphi \wedge \langle\langle A\rangle\rangle\bigcirc\langle\langle A\rangle\rangle\Box\varphi, \\ \langle\langle A\rangle\rangle\varphi\mathcal{U}\psi &\leftrightarrow \psi \vee (\varphi \wedge \langle\langle A\rangle\rangle\bigcirc\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi). \end{aligned}$$

Let $st_{\mathcal{E}}(\varphi)$ denote the set of states in which formula φ holds (in coalition effectivity model $\mathcal{E}$). From the observation above we obtain the following fixpoint characterizations of $\langle\langle A\rangle\rangle\Box\varphi$ and $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$:

$$\begin{aligned} st_{\mathcal{E}}(\langle\langle A\rangle\rangle\Box\varphi) &= \nu\mathbf{Z}.\left(st_{\mathcal{E}}(\varphi) \cap st_{\mathcal{E}}(\langle\langle A\rangle\rangle\bigcirc\mathbf{Z})\right), \\ st_{\mathcal{E}}(\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi) &= \mu\mathbf{Z}.\left(st_{\mathcal{E}}(\psi) \cup (st_{\mathcal{E}}(\varphi) \cap st_{\mathcal{E}}(\langle\langle A\rangle\rangle\bigcirc\mathbf{Z}))\right). \end{aligned}$$

Note that $st_{\mathcal{E}}(\langle\langle A\rangle\rangle\bigcirc \mathrm{Z})$ corresponds exactly to the set $pre_{\mathcal{E}}(A, \mathrm{Z})$, used within the presentation of the ATL model checking algorithm in (Alur et al., 2002). Function pre is employed there to go "one step back": it takes as input a coalition A and a set of states $\mathrm{Z} \subseteq Q$ and returns as output the set Z' of all states such that, when the system is in one of the states from Z', the agents A can cooperate and force the next state to be one of Z. We can use the function to obtain a clearer presentation of the semantics of ATL based on coalition effectivity models. Thus, let us finally define the semantics via the following clauses:

$$\begin{aligned} pre_{\mathcal{E}}(A, \mathbf{Z}) &= \{q \in \mathcal{E} \mid \mathbf{Z} \in E_q(A)\}, \\ st_{\mathcal{E}}(\langle\langle A\rangle\rangle\bigcirc\varphi) &= pre_{\mathcal{E}}(A, st_{\mathcal{E}}(\varphi)), \\ st_{\mathcal{E}}(\langle\langle A\rangle\rangle\Box\varphi) &= \nu\mathbf{Z}.\left(st_{\mathcal{E}}(\varphi) \cap pre_{\mathcal{E}}(A, \mathbf{Z})\right), \\ st_{\mathcal{E}}(\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi) &= \mu\mathbf{Z}.\left(st_{\mathcal{E}}(\psi) \cup (st_{\mathcal{E}}(\varphi) \cap pre_{\mathcal{E}}(A, \mathbf{Z}))\right) \\ \mathcal{E}, q \models \varphi \quad &\text{iff} \quad q \in st_{\mathcal{E}}(\varphi). \end{aligned}$$

2.7 Equivalence of the Different Semantics for ATL

In this section we compare the semantics for Alternating-time Temporal Logic, based on alternating transition systems and multi-player game models – and show their equivalence (in the sense that we can transform the models both ways while preserving satisfiability of ATL formulae). Further, we show that these semantics are both equivalent to the semantics based on coalition effectivity models.

The transformation from alternating transition systems to multi-player game models is easy: in fact, for every ATS, an equivalent MGM can be constructed via re-labeling transitions (see Section 2.7.1). Construction the other way round is more sophisticated: first, we observe that all multi-player game models obtained from alternating transition systems satisfy a special condition we call *convexity* (Section 2.7.1); then we show that for every convex MGM, an equivalent ATS can be obtained (Section 2.7.2). Finally, we demonstrate that for every arbitrary multi-player game model a convex MGM can be constructed that satisfies the same formulae of ATL (Section 2.7.3).

More analysis, showing how various structural properties of MGMs transfer to ATSs obtained through the transformations we propose (and vice versa), can be found in (Goranko and Jamroga, 2004).

2.7.1 From Alternating Transition Systems to MGMs

First, for every ATS $T = \langle \Pi, \mathbb{A}\mathrm{gt}, Q, \pi, \delta \rangle$ over a set of agents $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$ there is an equivalent MGM $M^T = \langle Q, \gamma^T, \pi \rangle$ where, for each $q \in Q$, the strategic game form $\gamma^T(q) = \langle \mathbb{A}\mathrm{gt}, \{\Sigma_a^q \mid a \in \mathbb{A}\mathrm{gt}\}, o_q, Q \rangle$ is defined in a very simple way:

- $\Sigma_a^q = \delta(q, a)$,
- $o_q(Q_{a_1}, ..., Q_{a_k}) = s$ where $\bigcap_{a_i \in \mathbb{A}\mathrm{gt}} Q_{a_i} = \{s\}$.

Example 2.8 Let us apply the transformation to the alternating transition system from Example 2.5. The resulting MGM is shown in Figure 2.7. The following proposition states that it satisfies the same ATL formulae as the original system. Note that – as T and M^T include the same set of states Q – the construction preserves validity of formulae (in the model), too. □

It is easy to observe that the transformation does not change the temporal nor strategic structure of the model – it only re-labels agents' choices. In this sense, M^T is isomorphic to T. The fact has an important consequence: the MGM we obtain through this transformation is equivalent to the original ATS in the context of ATL formulae.

Proposition 2.13 *For every alternating transition system T, a state q in it, and an* ATL *formula φ: $T, q \models \varphi$ iff $M^T, q \models \varphi$.*

The models M^T obtained as above share a specific property we call *convexity*, and define below. First, we need an auxiliary technical notion: a *fusion* of agents' choices.

Definition 2.10 *A* fusion *of n-tuples $(\alpha_1, ..., \alpha_n)$ and $(\beta_1, ..., \beta_n)$ is any n-tuple $(\gamma_1, ..., \gamma_n)$ where $\gamma_i \in \{\alpha_i, \beta_i\}$, $i = 1, ..., n$.*

Definition 2.11 *A strategic game form $\langle \mathbb{A}\mathrm{gt}, \{\Sigma_a \mid a \in \mathbb{A}\mathrm{gt}\}, Q, o \rangle$ is* convex *if:*

$$o(\sigma_{a_1}, ..., \sigma_{a_k}) = o(\tau_{a_1}, ..., \tau_{a_k}) = s \text{ implies } o(\varsigma_{a_1}, ..., \varsigma_{a_k}) = s \text{ for every fusion } (\varsigma_{a_1}, ..., \varsigma_{a_k}) \text{ of } (\sigma_{a_1}, ..., \sigma_{a_k}) \text{ and } (\tau_{a_1}, ..., \tau_{a_k}).$$

A multi-player game model $M = (Q, \gamma, \pi)$ is convex if $\gamma(q)$ is convex for every $q \in Q$.

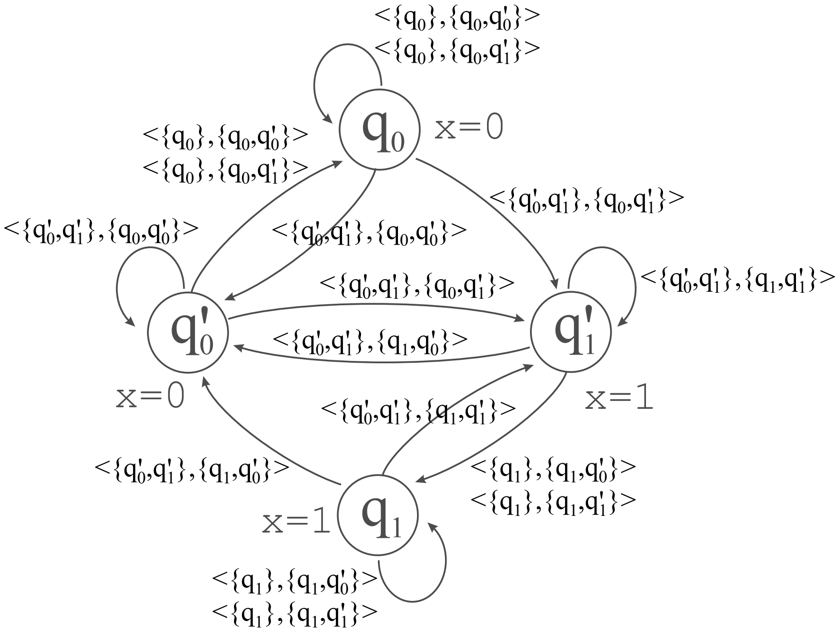


Figure 2.7: From ATS to a convex game structure: M^T for the system from Figure 2.5.

Proposition 2.14 *For every ATS T, the game model M^T is convex.*

Proof: Let M^T be defined as above. If $o_q(Q^1_{a_1}, ..., Q^1_{a_k}) = o_q(Q^2_{a_1}, ..., Q^2_{a_k}) = s$ then $s \in Q^j_a$ for each $j = 1, 2$ and $a \in \mathbb{A}\text{gt}$, therefore $\bigcap_{a \in \mathbb{A}\text{gt}} Q^{j_a}_a = \{s\}$ for any fusion $(Q^{j_1}_{a_1}, ..., Q^{j_k}_{a_k})$ of $(Q^1_{a_1}, ..., Q^1_{a_k})$ and $(Q^2_{a_1}, ..., Q^2_{a_k})$. □

There is an important subclass of convex game models, with a very simple characteristics:

Definition 2.12 *A strategic game form is* injective *if o is injective, i.e. assigns different outcome states to different tuples of choices. An MGM is injective if it contains only injective game forms.*

Proposition 2.15 *Every injective game model is convex.*

Note that the MGM from Figure 2.7 is convex, although it is not injective, so the reverse implication does not hold.

2.7.2 From Convex Multi-Player Game Models to Alternating Transition Systems

As it turns out, convexity is a sufficient condition if we want to re-label transitions from a multi-player game model back to an alternating transition system. Let $M = \langle Q, \gamma, \pi \rangle$

be a convex MGM over a set of propositions Π, where $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$, and let $\gamma(q) = \langle \mathbb{A}\mathrm{gt}, \{\Sigma_a^q \mid a \in \mathbb{A}\mathrm{gt}\}, Q, o_q\rangle$ for each $q \in Q$. We transform it to an ATS $T^M = \langle \Pi, \mathbb{A}\mathrm{gt}, Q, \pi, \delta^M\rangle$ with the transition function δ^M defined by

$$\delta^M(q, a) = \{Q_{\sigma_a} \mid \sigma_a \in \Sigma_a^q\},$$
$$Q_{\sigma_a} = \{o_q(\sigma_a, \sigma_{\mathbb{A}\mathrm{gt}\setminus\{a\}}) \mid \sigma_{\mathbb{A}\mathrm{gt}\setminus\{a\}} = \langle \sigma_{b_1}, ..., \sigma_{b_{k-1}}\rangle, b_i \neq a, \sigma_{b_i} \in \Sigma_{b_i}^q\}.$$

Thus, Q_{σ_a} is the set of states to which a transition may be effected from q while agent a has chosen to execute σ_a. Moreover, $\delta^M(q, a)$ simply collects all such sets. For purely technical reasons we regard these $\delta^M(q, a)$ as *indexed families* i.e. even if some Q_{σ_1} and Q_{σ_2} are set-theoretically equal, they are considered different as long as $\sigma_1 \neq \sigma_2$. By convexity of $\gamma(q)$ it is easy to verify that $\bigcap_{a\in\mathbb{A}\mathrm{gt}} Q_{\sigma_a} = \{o_q(\sigma_{a_1}, ..., \sigma_{a_k})\}$ for every tuple $(Q_{\sigma_{a_1}}, ..., Q_{\sigma_{a_k}}) \in \delta^M(q, a_1)\times ...\times\delta^M(q, a_k)$. Furthermore, the following proposition holds.

Proposition 2.16 *For every convex MGM M, a state q in it, and an* ATL *formula φ, $M, q \models \varphi$ iff $T^M, q \models \varphi$.*

Note that the above construction transforms the multi-player game model from Figure 2.7 exactly back to the ATS from Figure 2.5.

2.7.3 Equivalence between the Semantics for ATL Based on ATS and MGM

So far we have shown how to transform alternating transition systems to convex multi-player game models, and vice versa. Unfortunately, not every MGM is convex. However, for every MGM we can construct a convex multi-player game model that satisfies the same formulae of ATL. This can be done by creating distinct copies of the original states for different incoming transitions, and thus "storing" the knowledge of the previous state and the most recent choices from the agents in the new states. Since the actual choices are present in the label of the resulting state, the new transition function is obviously injective. It is also easy to observe that the construction given below preserves not only satisfiability, but also validity of formulae (in the model).

Proposition 2.17 *For every MGM $M = \langle Q, \gamma, \pi\rangle$ there is an injective (and hence convex) MGM $M' = \langle Q', \gamma', \pi'\rangle$ which satisfies the same formulae of* ATL.

Proof: For every $\gamma(q) = \langle \mathbb{A}\mathrm{gt}, \{\Sigma_a^q \mid a \in \mathbb{A}\mathrm{gt}\}, Q, o_q\rangle$ we define $Q_q = \{q\} \times \prod_{a\in\mathbb{A}\mathrm{gt}} \Sigma_a^q$ and let $Q' = Q \cup \bigcup_{q\in Q} Q_q$. Now we define γ' as follows:

- for $q \in Q$, we define $\gamma'(q) = \langle \mathbb{A}\mathrm{gt}, \{\Sigma_a^q \mid a \in \mathbb{A}\mathrm{gt}\}, O^q, Q'\rangle$, and $O^q(\sigma_{a_1}, ..., \sigma_{a_k}) = \langle q, \sigma_{a_1}, ..., \sigma_{a_k}\rangle$;
- for $\sigma = \langle q, \sigma_{a_1}, ..., \sigma_{a_k}\rangle \in Q_q$, and $s = o_q(\sigma_{a_1}, ..., \sigma_{a_k})$, we define $\gamma'(\sigma) = \gamma'(s)$;
- finally, $\pi'(q) = \pi(q)$ for $q \in Q$, and $\pi'(\langle q, \sigma_{a_1}, ..., \sigma_{a_k}\rangle) = \pi(o_q(\sigma_{a_1}, ..., \sigma_{a_k}))$ for $\langle q, \sigma_{a_1}, ..., \sigma_{a_k}\rangle \in Q_q$.

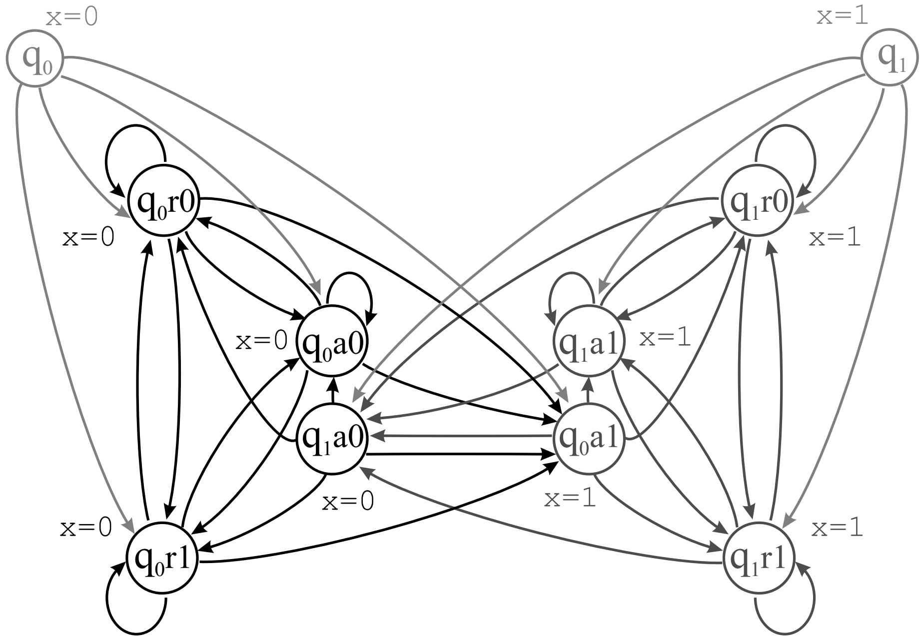


Figure 2.8: Construction of a convex multi-player game model equivalent to the MGM from Figure 2.3.

The model M' is injective and it can be proved by a straightforward induction that for every ATL formula φ:

- $M', q \models \varphi$ iff $M, q \models \varphi$ for $q \in Q$, and
- $M', \langle \sigma_{a_1}, ..., \sigma_{a_k} \rangle \models \varphi$ iff $M, o_q(\sigma_{a_1}, ..., \sigma_{a_k}) \models \varphi$ for $\langle \sigma_{a_1}, ..., \sigma_{a_k} \rangle \in Q_q$.

□

Thus, the restriction of the semantics of ATL to the class of injective (and hence to convex, as well) MGMs does not introduce new validities – and we obtain the following result.

Corollary 2.18 *For every* ATL *formula φ the following statements are equivalent:*

1. *φ is valid in all alternating transition systems.*
2. *φ is valid in all multi-player game models.*

Remark 2.19 *The above construction preserves validity and satisfiability of* ATL* *formulae, too (Jamroga and van der Hoek, 2003).*

δ	$q_0,\ q_0r0,\ q_0r1,\ q_0a0,\ q_1a0$	$q_1,\ q_1r0,\ q_1r1,\ q_1a1,\ q_0a1$
s	$\{\{q_0r0, q_0r1\}, \{q_0a0, q_0a1\}\}$	$\{\{q_1r0, q_1r1\}, \{q_1a0, q_1a1\}\}$
c	$\{\{q_0r0, q_0a0\}, \{q_0r1, q_0a1\}\}$	$\{\{q_1r0, q_1a0\}, \{q_1r1, q_1a1\}\}$

Figure 2.9: ATS-style transition function for the convex game model from Figure 2.8.

Example 2.9 We can apply the construction to the controller from Example 2.3, and obtain a convex MGM equivalent to the original one in the context of ATL. The result is displayed in Figure 2.8. The labels for the transitions can be easily deduced from their target states. Re-writing the game model into an isomorphic ATS, according to the construction from Section 2.7.2 (see Figure 2.9), completes the transformation from an arbitrary multi-player game model to an alternating transition system for which the same ATL formulae hold. □

2.7.4 ATS or MGM?

Alur stated that the authors of ATL switched from alternating transition systems to concurrent game structures mostly to improve understandability of the logic and clarity of the presentation.[4] Indeed, identifying actions with their outcomes may make things somewhat artificial and unnecessarily complicated. In particular, we find the convexity condition which ATSs impose too strong and unjustified in many situations. For instance, consider the following variation of the 'Chicken' game: two cars running against each other on a country road and each of the drivers, seeing the other car, can take any of the actions: *"drive straight"*, *"swerve to the left"* and *"swerve to the right"*. Each of the combined actions for the two drivers: ⟨*drive straight, swerve to the left*⟩ and ⟨*swerve to the right, drive straight*⟩ leads to a non-collision outcome, while each of their fusions ⟨*drive straight, drive straight*⟩ and ⟨*swerve to the left, swerve to the right*⟩ leads to a collision. Likewise, in the "Coordinated Attack" scenario (Fagin et al., 1995) any non-coordinated one-sided attack leads to defeat, while the coordinated attack of both armies, which is a fusion of these, leads to a victory. Thus, the definition of outcome function in coalition games is more general and flexible in our opinion.

Let us consider the system from Example 2.3 again. The multi-player game model (or concurrent game structure) from Figure 2.3 looks natural and intuitive. Unfortunately, it can't be used in the version of ATL based on Alternating Transitions Systems. Speaking more formally, there is no isomorphic ATS for the multi-player game model from Figure 2.3 which fits the system description (or: in which the same properties hold as in the MGM). In consequence, an ATS modeling the same situation must be larger.

Proposition 2.20 *There exists no ATS with exactly two states, in which the same* ATL *formulae are valid as in the MGM from Example 2.3.*

[4] Private communication.

Proof: Suppose that such an ATS exists. Let us have a look at $\delta(q_0, s)$ first. Suppose that $\{q_0\} \notin \delta(q_0, s)$. In consequence, $\{q_0, q_1\}$ must be in $\delta(q_0, s)$, otherwise no transition from q_0 to q_1 is possible. Let's consider possible c's choices at q_0: either $\{q_0, q_1\} \in \delta(q_0, c)$ (but: $\{q_0, q_1\} \cap \{q_0, q_1\}$ isn't a singleton, so such a transition function isn't valid) or $\{q_1\} \in \delta(q_0, c)$ (but then: $q_0 \models \langle\langle c \rangle\rangle \bigcirc \texttt{x=1}$ which doesn't fit the informal description of the system) or $\{q_0\} \in \delta(q_0, c)$ (but then: $q_0 \models \neg\langle\langle s, c \rangle\rangle \bigcirc \texttt{x=1}$ which doesn't fit either). Thus, $\{q_0\} \in \delta(q_0, s)$ (intuitively: the server should have a choice to "enforce no change" with deterministic outcome of $\{q_0\}$).

Now, for all $Q' \in \delta(q_0, c)$, q_0 must be in Q' because $\{q_0\} \cap Q'$ cannot be empty. Thus $\{q_1\} \notin \delta(q_0, c)$, and if we want to make the transition from q_0 to q_1 possible at all then $\{q_0, q_1\} \in \delta(q_0, c)$. Now $\{q_0, q_1\} \notin \delta(q_0, s)$ because $\{q_0, q_1\} \cap \{q_0, q_1\}$ isn't a singleton, so $\{q_1\} \in \delta(q_0, s)$ – otherwise the system still never proceeds from q_0 to q_1. In consequence, $\{q_0\} \notin \delta(q_0, c)$, because $\{q_1\} \cap \{q_0\}$ isn't a singleton either. The resulting transition function for q_0 is: $\delta(q_0, s) = \{\{q_0\}, \{q_1\}\}$, and $\delta(q_0, c) = \{\{q_0, q_1\}\}$. Unfortunately, it is easy to show that $q_0 \models \langle\langle s \rangle\rangle \bigcirc \texttt{x=1}$ for this model, and this is obviously wrong with respect to the original description of the system. □

This does not necessarily mean that no ATS can be made up for this problem, having added some extra states and transitions. In fact, for the alternating transition system from Figure 2.5, we have that:

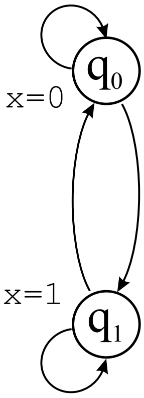
- $q_0 \models \neg\langle\langle s \rangle\rangle \bigcirc \texttt{x=1}$,
- $q_0 \models \langle\langle s \rangle\rangle \bigcirc \texttt{x=0}$, and so on.

The states reflect the value of x *and* the last choices made by the agents: q_0 is for "x=0 by s's force", q_0' for "x=0 by c's request" etc. This kind of construction has been generalized in Section 2.7 to prove equivalence of both semantics. The above examples, however, show that correct alternating transition systems are more difficult to come up with directly than multi-player game models, and usually they are more complex, too. This should be especially evident when we consider modeling and verifying open systems. Suppose we need to add another client process to the ATS from Example 2.5. It would be hard to extend the existing transition function in a straightforward way so that it still satisfies the formal requirements (i.e. so that all the intersections of choices are singletons). Designing a completely new ATS is probably an easier solution.

Another interesting issue is extendibility of the formalisms. Game models incorporate explicit labels for agents' choices – therefore the labels can be used, for instance, to restrict the set of valid strategies under uncertainty (cf. Chapter 4).

2.7.5 Coalition Effectivity Models as Equivalent Alternative Semantics for ATL

Effectivity functions and coalition effectivity models were introduced in Section 2.6, including a characterization of these effectivity functions which describe abilities of agents and their coalitions in actual strategic game forms (playable effectivity functions, Theorem 2.11). We are going to extend the result to correspondence between multi-player game models and standard coalition effectivity models (i.e. the coalition effectivity models that contain only playable effectivity functions).



$\varnothing$	$\{s\}$	$\{c\}$	$\{s,c\}$
$\{\{q_0,q_1\}\}$	$\{\{q_0\},\{q_0,q_1\}\}$	$\{\{q_0\},\{q_0,q_1\}\}$	$\{\{q_0\},\{q_1\},\{q_0,q_1\}\}$

$\varnothing$	$\{s\}$	$\{c\}$	$\{s,c\}$
$\{\{q_0,q_1\}\}$	$\{\{q_1\},\{q_0,q_1\}\}$	$\{\{q_1\},\{q_0,q_1\}\}$	$\{\{q_0\},\{q_1\},\{q_0,q_1\}\}$

Figure 2.10: Coalition effectivity model for the variable client/server system

Every MGM $M = \langle Q, \gamma, \pi \rangle$ for the set of players $\mathbb{A}\mathrm{gt}$ corresponds to a CEM $\mathcal{E}^M = \langle \mathbb{A}\mathrm{gt}, Q, E^M, \pi \rangle$, where for every $q \in Q$, $X \subseteq Q$ and $A \subseteq \mathbb{A}\mathrm{gt}$, we have

$$X \in E_q^M(A) \text{ iff } \exists \sigma_A \, \forall \sigma_{\mathbb{A}\mathrm{gt} \setminus A} \, \exists s \in X \; o(\sigma_A, \sigma_{\mathbb{A}\mathrm{gt}\setminus A}) = s.$$

The choices refer to the strategic game form $\gamma(q)$. Conversely, by Theorem 2.11, for every standard coalition effectivity model $\mathcal{E}$ there is a multi-player game model M such that $\mathcal{E}$ is equivalent to $\mathcal{E}^M$. Again, by a straightforward induction on formulae, we obtain:

Proposition 2.21 *For every MGM M, a state q in it, and an* ATL *formula φ, we have $M, q \models \varphi$ iff $\mathcal{E}^M, q \models \varphi$.*

Example 2.10 Let M be the multi-player game model from Example 2.3 (variable client/server system). Coalition effectivity model $\mathcal{E}^M$ is presented in Figure 2.10. □

By Proposition 2.21, Corollary 2.18 and Corollary 2.12, we eventually obtain:

Theorem 2.22 *For every* ATL *formula φ the following are equivalent:*

1. *φ is valid in all alternating transition systems,*
2. *φ is valid in all multi-player game models,*
3. *φ is valid in all standard coalition effectivity models.*

Thus, the semantics of ATL based on alternating transition systems, multi-player game models, and standard coalition effectivity models are equivalent. We note that, while the former two semantics are more concrete and natural, they are mathematically less elegant and suitable for formal reasoning about ATL, while the semantics based on coalition effectivity models is essentially a monotone neighborhood semantics for multi-modal logics. The combination of these semantics was used in (Goranko and van Drimmelen, 2003) to establish a complete axiomatization of ATL.

2.7.6 Relevance of the Results

We have presented a comparative study of several variants of ATL and CL, and demonstrated their relationship. One obvious conclusion from the study is that – while referring to coalitional games with no uncertainty – ATL can be used instead of CL without any loss of generality. Moreover, one can choose the semantics (multi-player game models, alternative transition systems, coalition effectivity models) he or she finds most suitable for the intended application.

Still, it is worth pointing out that ATL and CL differ in their motivations and agendas, and hence they can benefit from many ideas and results, both technical and conceptual, borrowing them from each other. Indeed, ATL has already benefited from being related to coalitional games, as concurrent game structures provide a more general (and natural) semantics than alternating transition systems. Moreover, coalition effectivity models are mathematically simpler and more elegant, and provide technically handier semantics, essentially based on neighborhood semantics for non-normal modal logics (Parikh, 1985; Pauly, 2000). Furthermore, the pure game-theoretical perspective of coalition logics can offer new ideas to the framework of open multi-agent systems and computations formalized by ATL. For instance, fundamental concepts in game theory, such as *preference relations between outcomes,* and *Nash equilibria* have their counterparts in concurrent game structures (and, more importantly, in alternating-time logics) which are unexplored yet.

On the other hand, the language and framework of ATL has widened the perspective on coalitional games and logics, providing a richer and more flexible vocabulary to talk about abilities of agents and their coalitions. The *alternating refinement relations* (Alur et al., 1998b) offer an appropriate notion of bisimulation between ATSs and thus can suggest an answer to the question "*When are two coalition games equivalent?*".[5] Also, a number of technical results on expressiveness and complexity, as well as realizability and model-checking methods from (Alur et al., 2002, 1998b) can be transferred to coalition games and logics. And there are some specific aspects of computations in open systems, such as *controllability* and *fairness constraints*, which have not been explored in the light of coalition games.

There were a few attempts to generalize ATL by including imperfect information in its framework: ATL *with incomplete information* in (Alur et al., 2002), ATEL, ATOL, ATEL-R* etc. It can be interesting to see how these attempts carry over to the framework of CL. Also, stronger languages like ATL* and alternating-time μ-calculus can provide more expressive tools for reasoning about coalition games.

2.8 Multi-Agent Planning with ATL

Planning in an environment inhabited by multiple agents poses a number of important problems. First of all, the relationship between the agents' goals must be determined in order to evaluate plans. There are roughly 3 possibilities (Sen and Weiss, 1999; Bui and Jamroga, 2003):

[5] Cf. the paper "When are two games the same" in (van Benthem, 2000).

1. collaborative agents who have exactly the same goals and can collaborate to bring about them,
2. adversarial agents: the goal of the other agents is *not to let* "our" agent fulfill his goals,
3. indifferent agents (or rather agents with independent policies) that cover all the middle ground between both extremes.

As ATL strategies are in fact conditional plans, one may imagine using ATL for the purpose of planning as long as we are able to automatically *find* a strategy that brings about a specified task. For example, a plan for agent a to eventually enforce a winning position (i.e. a state in which proposition win holds) can be found in the form of a strategy f_a that makes formula $\langle\langle a \rangle\rangle \Diamond \mathsf{win}$ true in the specified model.

Planning with ATL covers only cases (1) and (2). The collaborative situation arises when we put the agents in question in the same team A, and ask whether they achieve goal Φ via the formula $\langle\langle A \rangle\rangle \Phi$. Note, however, that the agents within a team are assumed to *fully* cooperate – as if they were one (collective) player. Therefore one must be careful to define the model so that it indeed represents the intended problem domain. For example, if we want the agents to negotiate their strategies by following a negotiation protocol, represented explicitly in the model, we should rule out all agents' choices that are inconsistent with the requirement.

The assumed interaction between the team and the rest of agents from $\mathbb{A}\mathrm{gt}$, on the other hand, is clearly adversarial. This reflects a bias of classical game theory: we want our agent to play safe; we want him to be protected against the worst line of events. Note that if there is no infallible plan for A to achieve Φ, then no plan will be generated for $\langle\langle A \rangle\rangle \Phi$ at all. Since a situation when no plan is generated is not acceptable from the planning agent's perspective, it seems one of the most serious drawbacks of the approach to planning we propose and investigate here.

A number of simplifying assumptions underlies the "planning with ATL approach":

- the agents have complete knowledge of the situation (no uncertainty, no probabilistic beliefs),
- the agents have complete knowledge of the outcomes of every combination of actions from all the agents (i.e. every complete tuple of choices),
- the outcome of every such tuple is deterministic (there are no probabilistic actions or actions with uncertain outcome),
- the time is discrete, and the agents act synchronously,
- the goals of every agent are public. Note that the goals themselves are specified by the input ATL formula we are processing.

2.8.1 Strategic Planning as Model Checking

Model checking is an interesting idea that emerged from the research on logic in computer science. The model checking problem asks whether a particular formula φ holds in a particular model M, which is often more interesting than *satisfiability checking* (i.e. looking for a model M in which φ holds) or *theorem proving* (i.e. proving that φ follows from some set of axioms). In many cases the designer can come up with a precise model of the system behavior (e.g. a graph with all the actions that may be effected), only the model is too large to check on the fly whether it fulfills the design objectives. Model checking seems especially useful in the case of dynamic or temporal logics, whose models can be interpreted as game models, transition systems, control flow charts, data flow charts etc. Moreover, model checking turns out to be relatively cheap in computational terms, while satisfiability checking is often intractable or even undecidable.

It has been already proposed that the model checking of computation tree logic (CTL) formulae can be used for generating plans in deterministic as well as non-deterministic domains (Giunchiglia and Traverso, 1999; Pistore and Traverso, 2001). Alternating-time temporal logic ATL is an extension of CTL that includes notions of agents, their abilities and strategies (conditional plans) explicitly in its models. Thus, ATL seems even better suited for planning, especially in multi-agent systems, which was already suggested in (van der Hoek and Wooldridge, 2002). In this section, we introduce a simple adaptation of the ATL model checking algorithm from (Alur et al., 2002) that – besides checking if a goal can be achieved – returns also an appropriate strategy to achieve it. We point out that this algorithm generalizes the well-known search algorithm of minimaxing, and that ATL models generalize turn-based transition trees from game theory. The section ends with some suggestions that the contribution can be bilateral, and that more game theory concepts can contribute to modal logic-based models and algorithms for multi-agent systems.

2.8.2 Planning Algorithm

In this section, a simple modification of the ATL model checking algorithm (Alur et al., 2002) is proposed, as shown in Figure 2.11. Function pre is defined as a special kind of the "weakest precondition" operator, and is used here to go "one step back" while constructing a plan for some coalition of agents. More precisely, $pre(A, Q_1)$ takes as input a coalition A and a set of states $Q_1 \subseteq Q$ and returns as output the set Q_2 of all states q such that agents A can cooperate in q and force the next state to be one of Q_1. Moreover, for every such a state, $pre(A, Q_1)$ returns also a collective choice for A that can be executed to enforce a transition to Q_1.

Function $states(P)$ returns all the states for which plan P is defined. $P_1 \oplus P_2$ refers to augmenting plan P_1 with all new subplans that can be found in P_2; finally $P|_{Q_1}$ denotes plan P restricted to the states from Q_1 only. More formally:

- $pre(A, Q_1) = \{\langle q, \sigma_A \rangle \mid \forall \sigma_{\Sigma \setminus A} \delta(q, \sigma_A, \sigma_{\Sigma \setminus A}) \in Q_1\}$;
- $states(P) = \{q \in Q \mid \exists_\sigma \langle q, \sigma \rangle \in P\}$;
- $P_1 \oplus P_2 = P_1 \cup \{\langle q, \sigma \rangle \in P_2 \mid q \notin states(P_1)\}$;
- $P|_{Q_1} = \{\langle q, \sigma \rangle \in P \mid q \in Q_1\}$.

function $plan(\varphi)$**.**
Returns a subset of Q for which formula φ holds, together with a (conditional) plan to achieve φ. The plan is sought within the context of concurrent game structure $S = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, o \rangle$.

case $\varphi \in \Pi$ **:** return $\{\langle q, - \rangle \mid \varphi \in \pi(q)\}$
case $\varphi = \neg\psi$ **:** $P_1 := plan(\psi)$;
  return $\{\langle q, - \rangle \mid q \notin states(P_1)\}$
case $\varphi = \psi_1 \vee \psi_2$ **:**
  $P_1 := plan(\psi_1)$; $P_2 := plan(\psi_2)$;
  return $\{\langle q, - \rangle \mid q \in states(P_1) \cup states(P_2)\}$
case $\varphi = \langle\langle A \rangle\rangle \bigcirc \psi$ **:** return $pre(A, states(plan(\psi)))$
case $\varphi = \langle\langle A \rangle\rangle \Box \psi$ **:**
  $P_1 := plan(\top)$; $P_2 := plan(\psi)$; $Q_3 := states(P_2)$;
  while $states(P_1) \not\subseteq states(P_2)$
  do $P_1 := P_2|_{states(P_1)}$; $P_2 := pre(A, states(P_1))|_{Q_3}$ **od**;
  return $P_2|_{states(P_1)}$
case $\varphi = \langle\langle A \rangle\rangle \psi_1 \mathcal{U} \psi_2$ **:**
  $P_1 := \emptyset$; $Q_3 := states(plan(\psi_1))$; $P_2 := plan(\top)|_{states(plan(\psi_2))}$;
  while $states(P_2) \not\subseteq states(P_1)$
  do $P_1 := P_1 \oplus P_2$; $P_2 := pre(A, states(P_1))|_{Q_3}$ **od**;
  return P_1
end case

Figure 2.11: Adapted model checking algorithm for ATL formulae. Cases for $\psi_1 \vee \psi_2$ and $\langle\langle A \rangle\rangle \Diamond \psi$ are omitted, because the first can be re-written as $\neg(\neg\psi_1 \vee \neg\psi_2)$, and the latter as $\langle\langle A \rangle\rangle \top \mathcal{U} \psi$.

Proposition 2.23 *The algorithm terminates in time $O(ml)$, where m is the number of transitions in the concurrent game structure S, and l is the length of formula φ.*

Proof: The proposition follows directly from the complexity proofs for ATL model checking (Alur et al., 2002). □

Remark 2.24 *Note that the algorithm returns a (non-empty) plan only if the outmost operator of the checked formula is a cooperation modality (i.e. it specifies explicitly* who *is to execute the plan and* what *is the objective). In consequence, our approach to negation is* not *constructive: for $\neg\langle\langle A \rangle\rangle \Phi$, the algorithm will not return a strategy for the rest of agents to actually avoid Φ. Why? Because $\neg\langle\langle A \rangle\rangle \Phi$ does not imply that such a strategy exists.*

Similar remark applies to alternative, conjunction, and nesting of strategic formulae. This approach is more natural than it seems at the first glance – even if the subformulae refer to the same set of agents for whom plans are needed. Consider, for instance, the transition system from Figure 2.12, and suppose that there is only one agent a in the system, who executes the transitions. Formula $\langle\langle a \rangle\rangle \Box \mathsf{start} \wedge \langle\langle a \rangle\rangle \Diamond \mathsf{halt}$

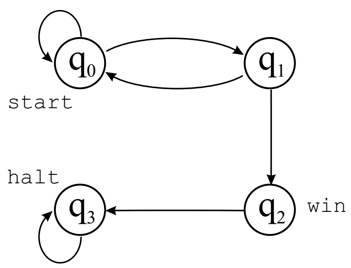


Figure 2.12: Example transition system for a single agent

is obviously true in q_0; however, it is hard to see what plan should be generated in this case. True, a has a plan to remain in q_0 for ever, and he has a plan to halt the system eventually, but these are different *plans and cannot be combined. Similarly, $\langle\langle a\rangle\rangle\Box\langle\langle a\rangle\rangle\Diamond\mathsf{win}$ holds in q_0, but it does not mean that a has a plan to win infinitely many times. He can always see a way to win; however, if he chooses that way, he will be unable to win again.*

2.8.3 Rocket Example

As an example, consider a modified version of the Simple Rocket Domain from (Blum and Furst, 1997). The task is to ensure that a cargo eventually arrives in Paris (proposition atCP); there are three agents with different capabilities who can be involved, and a single rocket that can be used to accomplish the task. Initially, the cargo may be in Paris, at the London airport (atCL) or it may lie inside the rocket (inCR). Accordingly, the rocket can be moved between London (atRL) and Paris (atRP).

There are three agents: x who can load the cargo, unload it, or move the rocket; y who can unload the cargo or move the rocket, and z who can load the cargo or supply the rocket with fuel (action *fuel*). Every agent can also decide to do nothing at a particular moment (the nop – "no-operation" action). The agents act simultaneously. The "moving" action has the highest priority (so, if one agent tries to move the rocket and another one wants to, say, load the cargo, then only the moving is executed). "Loading" is effected when the rocket does not move and more agents try to load than to unload. "Unloading" works in a similar way (in a sense, the agents "vote" whether the cargo should be loaded or unloaded). If the number of agents trying to load and unload is the same, then the cargo remains where it was. Finally, "fueling" can be accomplished alone or in parallel with loading or unloading. The rocket can move only if it has some fuel (fuelOK), and the fuel must be refilled after each flight. We assume that all the agents move with the rocket when it flies to another place. The concurrent game structure for the domain is shown in Figure 2.13.

$$plan(\langle\langle x\rangle\rangle\Diamond\mathsf{atCP}) = \{\ \langle 9,-\rangle,\langle 10,-\rangle,\langle 11,-\rangle,\langle 12,-\rangle\ \} \quad (2.1)$$

$$plan(\langle\langle x,y\rangle\rangle\Diamond\mathsf{atCP}) = \{\ \langle 2, x\!:\!load\cdot y\!:\!nop\rangle,\langle 6, x\!:\!move\cdot y\!:\!nop\rangle, \quad (2.2)$$

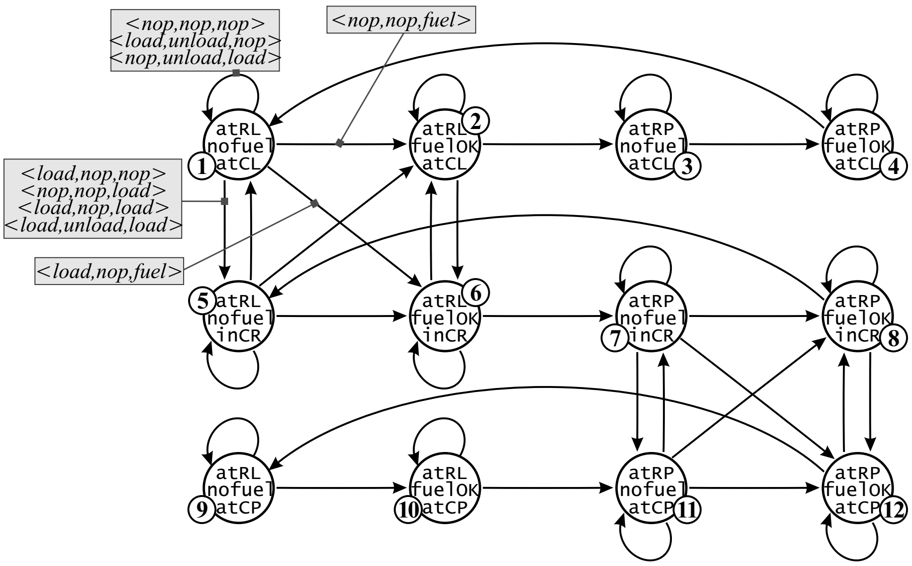


Figure 2.13: A version of the Simple Rocket Domain. States of the system are labeled with natural numbers. All the transitions for state 1 (the cargo and the rocket are in London, no fuel in the rocket) are labeled. Output of agents' choices for other states is analogous.

$$
\begin{array}{rcl}
 & & \langle 7, x\!:\!unload\cdot y\!:\!unload\rangle, \langle 8, x\!:\!unload\cdot y\!:\!unload\rangle, \\
 & & \langle 9, -\rangle, \langle 10, -\rangle, \langle 11, -\rangle, \langle 12, -\rangle \ \} \\
plan(\langle\langle x, z\rangle\rangle \Diamond \mathsf{atCP}) & = & \{ \ \langle 1, x\!:\!load\cdot z\!:\!load\rangle, \langle 2, x\!:\!load\cdot z\!:\!load\rangle, \qquad (2.3) \\
 & & \langle 3, x\!:\!nop\cdot z\!:\!fuel\rangle, \langle 4, x\!:\!move\cdot z\!:\!nop\rangle, \\
 & & \langle 5, x\!:\!load\cdot z\!:\!fuel\rangle, \langle 6, x\!:\!move\cdot z\!:\!nop\rangle, \\
 & & \langle 7, x\!:\!unload\cdot z\!:\!nop\rangle, \langle 8, x\!:\!unload\cdot z\!:\!nop\rangle, \\
 & & \langle 9, -\rangle, \langle 10, -\rangle, \langle 11, -\rangle, \langle 12, -\rangle \ \}
\end{array}
$$

Plans to eventually achieve atCP – for x alone, x with y, and x with z, respectively – are shown above. In the first case, x cannot guarantee to deliver the cargo to Paris (unless the cargo already *is* there), because y and z may prevent him from unloading the goods (clause 2.1). The coalition of x and y is more competent: they can, for instance, deliver the cargo from London if only there is fuel in the rocket (clause 2.2). However, they have no infallible plan for the most natural case when 1 is the initial state. Finally, $\{x, z\}$ have an effective plan for any initial situation (clause 2.3).

2.8.4 Minimaxing as Model Checking

It is easy to see that the algorithm from Figure 2.11 can be used for emulating the well known search algorithm of minimaxing. To find the best plan for coalition A,

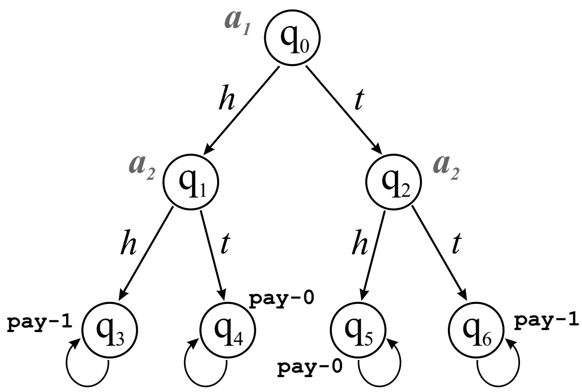


Figure 2.14: Multi-player game model for the matching pennies game

we should label the final positions with the payoff values pay-1, pay-2, ..., then check which $plan(\langle\langle A\rangle\rangle\Diamond \mathsf{pay\text{-}i})$ returns a decision for the initial state, and pick the one for maximal pay-i. The resulting procedure is still linear in the number of states, transitions and different payoff values. Note that the algorithm proposed here is more general than the original minimaxing: the latter can be applied only to finite turn-based game trees (i.e. systems in which the number of states is finite, there are no cycles, and players cannot act simultaneously), while the model checking-based approach deals also with models in which players act in parallel, and with infinite trees that can be generated by a finite transition system.

Example 2.11 Consider the perfect information variant of the *matching pennies* game from Example 2.1. Figure 2.14 shows a model representing this game. Now, minimaxing for player a_1 is equivalent to the execution of $plan(\langle\langle a_1\rangle\rangle\Diamond \mathsf{pay\text{-}1})$. We can also execute $plan(\langle\langle a_1\rangle\rangle\Diamond \mathsf{pay\text{-}0})$ to make sure that the agent is going to get *any* payoff along every course of action. □

Let us also observe that the planning algorithm, proposed in this section, looks for a plan that must be successful against every line of events – hence the algorithm generalizes minimaxing in zero-sum (i.e. strictly competitive) games. It can be interesting to model the non-competitive case within the scope of ATL as well: while checking $\langle\langle A\rangle\rangle\varphi$, the opponents $\mathbb{A}\mathrm{gt} \setminus A$ may be assumed different goals than just to prevent A from achieving φ. Then, assuming optimal play from $\mathbb{A}\mathrm{gt} \setminus A$, we can ask whether A have a strategy to enforce φ provided that $\mathbb{A}\mathrm{gt} \setminus A$ intend (or desire) to bring about ψ.

2.8.5 Further Research: Exploiting the Parallel between Model Checking and Minimaxing

We have proposed a simple adaptation of ATL model checking from (Alur et al., 2002). The algorithm looks for infallible conditional plans to achieve objectives that can be defined via ATL formulae. The algorithm generalizes minimaxing in zero-sum games, extending its scope to (possibly infinite) games in which the agents can act in parallel.

It seems that the link between model checking and minimaxing can be exploited to enrich the framework of ATL, too. First (as already mentioned in the previous section), ATL might be extended so that it can be used to model non-competitive games. Next, efficient pruning techniques exist for classical minimaxing – it may be interesting to transfer them to ATL model checking. Moreover, game theory has developed more sophisticated frameworks, like games with incomplete information and games with probabilistic outcomes, including the discussion on best defense criteria for such games (Frank, 1996; Frank and Basin, 1998; Jamroga, 2001a). Investigation of similar concepts in the context of ATL can prove worthwhile, and lead to new research questions, concerning phenomena like non-locality (Frank and Basin, 1998), and design of efficient suboptimal algorithms (Frank et al., 1998) in the scope of logics for multi-agent systems.

Chapter 3

Agents with Incomplete Information

SYNOPSIS. *So far, so good. Alternating-time Temporal Logic and Coalition Logic have been introduced, investigated and proven some properties. They even turned out to have a practical dimension, since one can use them for automatic planning in multi-agent systems. It seems a high time to introduce some uncertainty. Enter Alternating-time Temporal* Epistemic *Logic.*

Now we can express what we know when we do not know everything. But – do we really know what hides behind this attractive surface? Are we really certain? So far... yes. It seems so.

3.1 Introduction

Two important modal logics for multi-agent systems were studied in the previous chapter. Those logics, Alternating-time Temporal Logic by Alur et al. and Coalition Logic by Pauly, are appealing in many ways. Theoretically, they are decidable and axiomatizable, and they enjoy linear complexity of the model checking problem. Conceptually, they build upon a very intuitive body of notions. On one hand, they refer to moments in time and alternative courses of events and situations (where changes of situations result from simultaneous actions of all the involved agents). On the other hand, they build upon notions of agents, their teams, actions and strategies. From the practical standpoint, models of ATL and CL generalize labeled transition systems, control flow charts, game trees etc. – that have been used for a long time to obtain a formal description of computational systems as well as communities and organizations of human agents. Thus, linear model checking enables efficient verification of some interesting properties of concrete systems. Moreover, the "planning as model checking" paradigm, applied here, yields an algorithm that finds infallible plans for goals specified with formulae of ATL. As CL turned out to be subsumed by ATL, it is sufficient to focus on the latter, and regard CL as a sublanguage of ATL.

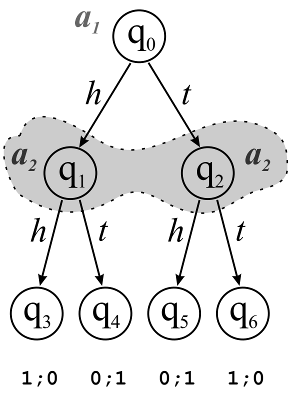
Mathematical logic seems to be rather out of fashion now, especially in the fields related to Artificial Intelligence and Cognitive Science. We believe – and have argued so in Section 2.1 – that formal approaches to multi-agents systems are still important. However, besides the inherent deficiencies of the formal logic-based approach to modeling and reasoning about the reality, the logics studied in Chapter 2 are unrealistic in one more respect: neither ATL nor CL refer in any way to agents' knowledge or beliefs about the actual situation. In particular, the agents' uncertainty about the real state of affairs cannot be expressed. An extension of ATL, called *Alternating-time Temporal Epistemic Logic* (ATEL), was introduced in (van der Hoek and Wooldridge, 2002) in order to enable reasoning about knowledge possessed by agents. Although the semantics for ATEL is still under debate (cf. Chapter 4), the original version of that logic is certainly worth investigating.

ATEL adds to ATL operators from epistemic logic: most notably, the individual knowledge operator $K_a\varphi$ ("agent a knows that φ"), but also operators E_A, C_A and D_A that refer to collective knowledge of teams. We begin with presenting the main ideas behind this extension. Next, we present three alternative semantics for ATEL, and point out that they are equivalent. Finally, we present an interpretation of ATEL into the more basic language of ATL. It turns out that, while extending ATL, ATEL can be embedded into the former in the sense that there is a translation of models and formulae of ATEL into ATL that preserves the satisfiability of formulae. This does not imply that logics like ATEL are redundant, of course – in fact, the way of expressing epistemic facts in ATL is technical, and the resulting formulae look rather unnatural. On the other hand, the interpretation we propose in Section 3.4 is not a purely technical exercise in formal logic. First, it presents knowledge as a special kind of strategic ability of agents. This perspective to knowledge was proposed in (van Otterloo et al., 2003), and we find it a powerful and insightful metaphor. Second, as the interpretation can be used for an efficient reduction of model checking from ATEL to ATL, it enables using the ATL-based planning algorithm from Section 2.8 for goals specified with ATEL formulae, too.

The epistemic and temporal layers of ATEL display much similarity to the well known BDI logic of Beliefs, Desires and Intentions, proposed in (Rao and Georgeff, 1991) and subsequent papers. We explore this issue in Section 3.5, and show that a similar interpretation in ATL (and even in CTL) can be defined for a propositional variant of BDI as well.

There is one more thing that may need some explanation – namely, the title of this chapter. Why *"Agents with Incomplete Information"*, and not "Agents with Knowledge" or "Agents with Beliefs", for instance? We would like to stress that epistemic properties become trivial when every agent has *complete* information about the current situation. Formally speaking, in such a case we have that $\varphi \leftrightarrow K_a\varphi$ (to be even more precise: formula $\varphi \leftrightarrow K_a\varphi$ is *valid*, i.e. it holds in every possible model and every state in it). Thus, the notion of knowledge does not introduce any interesting properties when all the agents have perfect and complete information about their environment of action. Talking about knowledge or beliefs – in our opinion – makes sense only if there is at least one entity who might *not* know everything.

Some of material presented in this chapter has already been published in (Goranko and Jamroga, 2004), a paper co-written with Valentin Goranko from the Rand Afrikaans University.



	q_1q_2h	q_1q_2t
q_0h	1;0	0;1
q_0t	0;1	1;0

Figure 3.1: Extensive and strategic form of the matching pennies game: a_1 does not show his coin before the end of the game.

3.2 Logic Meets Game Theory Continued

We have already presented (in Section 2.2) the concepts of strategic and extensive game forms, that can be used to model games with perfect information. The information available to agents is incomplete in many games, though. Classical game theory handles this kind of uncertainty through partitioning every player's nodes into so called information sets. An *information set* for player a is a set of states that are indistinguishable for a. Traditionally, information sets are defined only for the states in which a chooses the next step.[1] Now a strategy assigns choices to information sets rather than separate states, because players are supposed to choose the same move for all the situations they cannot distinguish.

Example 3.1 Let us go to the matching pennies game from Example 2.1 – but this time we will assume that a_1 does not show his coin to a_2 before the end of the game. In consequence, nodes q_1 and q_2 belong to the same information set of a_2, as shown in Figure 3.1. No player has a strategy that guarantees his win any more. □

Epistemic logic offers the notion of an *epistemic accessibility relation* that generalizes information sets, and introduces operators for talking about individual and collective knowledge. Section 3.3 describes them in more detail; a reader interested in a comprehensive exposition on epistemic logic can be also referred to the seminal book by Fagin, Halpern, Moses and Vardi (Fagin et al., 1995), or to (van der Hoek and Verbrugge, 2002) for a survey.

[1] A recent proposal (Bonnano, 2004) investigates also the case when information sets partition the *whole* game space for every agent.

3.3 ATEL: Adding Knowledge to Strategies and Time

Alternating-time Temporal Epistemic Logic ATEL (van der Hoek and Wooldridge, 2002, 2003a,b) enriches the picture with epistemic component. ATEL (ATEL*) adds to ATL (ATL*, respectively) operators for representing knowledge of agents and their teams:

- $K_a\varphi$, where a is an agent and φ is a formula of ATEL (ATEL* state formula, respectively);
- $C_A\varphi$, $E_A\varphi$ and $D_A\varphi$, where A is a set of agents and φ is a formula of ATEL (ATEL* state formula, respectively).

$K_a\varphi$ reads as "agent a knows that φ". Collective knowledge operators $E_A\varphi$, $C_A\varphi$, and $D_A\varphi$ refer to *"everybody knows"*, *common knowledge*, and *distributed knowledge* among the agents from A. Thus, $E_A\varphi$ means that every agent in A knows that φ holds. $C_A\varphi$ implies much more: the agents from A not only know that φ, but they also *know that they know* this, know that they know that they know, and so on. Distributed knowledge $D_A\varphi$ denotes a situation in which, if the agents could combine their individual knowledge together, they would be able to infer that φ holds.

The time complexity of model checking for ATEL (but not ATEL*) is still polynomial (van der Hoek and Wooldridge, 2002, 2003a).

Intuitively, ATEL should enable expressing various epistemic properties of agents under uncertainty:

1. $\langle\!\langle a \rangle\!\rangle \Diamond \varphi \rightarrow K_a\psi$
2. $K_b(c = s) \rightarrow \langle\!\langle b \rangle\!\rangle (\langle\!\langle b \rangle\!\rangle \bigcirc o)\,\mathcal{U} \neg(c = s)$
3. $d \rightarrow \langle\!\langle a \rangle\!\rangle \Diamond (K_a d \wedge \bigwedge_{a \neq b} \neg K_b d)$

The first two items are examples of so-called *knowledge pre-conditions*. The first of them intuitively says that agent a must know ψ in order to be able to bring about φ. The second expresses that if Bob (b) knows that the combination of the safe is s, then he is able to open it (o), as long as the combination remains unchanged. The third example refers to *Knowledge Games*, investigated in (van Ditmarsch, 2000) as a particular way of learning in multiagent systems. The aim of each player is to find out the actual distribution of cards (d) in a simple card game. The example specifies what it means for player a to have a winning strategy: it says that a can establish that eventually he will know the distribution, and all the other player will not know it.

3.3.1 AETS and Semantics of Epistemic Formulae

Models for ATEL are called *alternating epistemic transition systems* (AETS). They extend alternating transition systems with epistemic accessibility relations $\sim_1, ..., \sim_k \subseteq Q \times Q$ for modeling agents' uncertainty:

$$\mathcal{T} = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, \sim_{a_1}, ..., \sim_{a_k}, \delta \rangle.$$

These are assumed to be equivalence relations. Agent a's epistemic relation is meant to encode a's inability to distinguish between the (global) system states: $q \sim_a q'$ means that, while the system is in state q, agent a cannot really determine whether it is in q or q'. Then:

$$\mathcal{T}, q \models K_a\varphi \text{ iff for all } q' \text{ such that } q \sim_a q' \text{ we have } \mathcal{T}, q' \models \varphi$$

Remark 3.1 *Since the epistemic relations are required to be equivalences, the epistemic layer of* ATEL *refers indeed to agents'* knowledge *rather than* beliefs *in general. We suggest that this requirement can be relieved to allow* ATEL *for other kinds of beliefs as well. In particular, the interpretation of* ATEL *into* ATL *we propose in Section 3.4 does not assume any specific properties of the accessibility relations.*

Relations $\sim_A^E$, $\sim_A^C$ and $\sim_A^D$, used to model group epistemics, are derived from the individual accessibility relations of agents from A:

- first, $\sim_A^E$ is the union of the relations, i.e. $q \sim_A^E q'$ iff $q \sim_a q'$ for some $a \in A$. In other words, if everybody knows φ, then no agent may be unsure about the truth of it, and hence φ should be true in all the states that cannot be distinguished from the current state by even one member of the group.
- Next, $\sim_A^C$ is defined as the reflexive and transitive closure of $\sim_A^E$.
- Finally, $\sim_A^D$ is the intersection of all the $\sim_a$, $a \in A$: if any agent from A can distinguish q from q', then the whole group can distinguish the states, having combined their individual knowledge together.

The semantics of group knowledge can be defined as below:

$$\mathcal{T}, q \models E_A\varphi \quad \text{iff} \quad \text{for all } q' \text{ such that } q \sim_A^E q' \text{ we have } \mathcal{T}, q' \models \varphi$$
$$\mathcal{T}, q \models C_A\varphi \quad \text{iff} \quad \text{for all } q' \text{ such that } q \sim_A^C q' \text{ we have } \mathcal{T}, q' \models \varphi$$
$$\mathcal{T}, q \models D_A\varphi \quad \text{iff} \quad \text{for all } q' \text{ such that } q \sim_A^D q' \text{ we have } \mathcal{T}, q' \models \varphi$$

Remark 3.2 *Epistemic relation $\sim_A^C$ is usually defined as only the transitive closure of $\sim_A^E$ (van der Hoek and Verbrugge, 2002; van der Hoek and Wooldridge, 2002). The reflexivity of the closure changes nothing here, since all $\sim_a$ are defined to be reflexive themselves — except for $A = \emptyset$. And that is exactly why we add it: now $\sim_\emptyset^C$ can be used to describe having complete information.*

Example 3.2 Let us consider another variation of the variable/controller example: the client can try to add 1 or 2 to the current value of x now (the addition is modulo 3 in this case). Thus the operations available to c are: "$x := x + 1 \bmod 3$" and "$x := x+2 \bmod 3$". The server can still accept or reject the request from c (Figure 3.2). We depict the epistemic accessibility relations with dotted lines in the graph. In this case, the dotted lines show that c cannot distinguish being in state q_0 from being in state q_1, while s is not able to discriminate q_0 from q_2. Some formulae that are valid for this AETS (i.e. true in every state of the model) are shown below:

(A)

$\mathbb{A}\mathrm{gt} = \{s, c\}$ $Q = \{q_0, q_1, q_2\}$
$\Pi = \{\mathsf{x{=}0}, \mathsf{x{=}1}, \mathsf{x{=}2}\}$ $\pi(q_0) = \{\mathsf{x{=}0}\}$
$\pi(q_1) = \{\mathsf{x{=}1}\}$ $\pi(q_2) = \{\mathsf{x{=}2}\}$
$\delta(q_0, s) = \{\{q_0\}, \{q_1, q_2\}\}$
$\delta(q_1, s) = \{\{q_1\}, \{q_0, q_2\}\}$
$\delta(q_2, s) = \{\{q_2\}, \{q_0, q_1\}\}$
$\delta(q_0, c) = \{\{q_0, q_1\}, \{q_0, q_2\}\}$
$\delta(q_1, c) = \{\{q_1, q_0\}, \{q_1, q_2\}\}$
$\delta(q_2, c) = \{\{q_2, q_0\}, \{q_2, q_1\}\}$
$q_0 \sim_s q_0$ $q_0 \sim_s q_2$ $q_1 \sim_s q_1$
$q_2 \sim_s q_0$ $q_2 \sim_s q_2$
$q_0 \sim_c q_0$ $q_0 \sim_c q_1$ $q_1 \sim_c q_0$
$q_1 \sim_c q_1$ $q_2 \sim_c q_2$

(B)

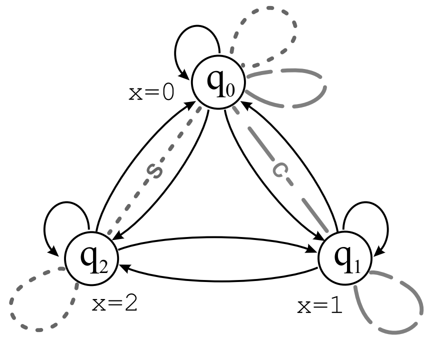


Figure 3.2: (A) An AETS for the modified controller/client problem. (B) The temporal and epistemic structure of the system: dotted lines display the epistemic accessibility relations for s and c.

1. $\mathsf{x{=}1} \rightarrow K_s\mathsf{x{=}1}$. In q_1 (which is the only state that satisfies $\mathsf{x{=}1}$), player s must consider only one possibility – namely, q_1 itself. Thus, s knows that $\mathsf{x{=}1}$ must hold now;

2. $\mathsf{x{=}2} \rightarrow E_{s,c}\neg\mathsf{x{=}1} \wedge \neg C_{s,c}\neg\mathsf{x{=}1}$: in q_2, both players can rule out the possibility that $\mathsf{x{=}1}$. On the other hand, they do not have common knowledge about it. The server knows that the current state is either q_2 or q_0, but it knows also that if q_0 is the case, then the client must consider q_1 possible. Thus, s cannot rule out a situation in which c believes that $\mathsf{x{=}1}$. In consequence, s and c know that $\neg\mathsf{x{=}1}$, but s does not know that c knows it, and common knowledge is not obtained;

3. $\mathsf{x{=}0} \rightarrow \langle\langle s\rangle\rangle\bigcirc\ \mathsf{x{=}0} \wedge \neg K_s\langle\langle s\rangle\rangle\bigcirc\ \mathsf{x{=}0}$. In q_0, the server can enforce that the system will be in q_0 in the next step (by choosing to reject the client's request). However, s does not know that he is able to do so, because he must consider q_2 as a possible state of affairs as well, and in q_2 such a strategy for s does not exist;

4. $\mathsf{x{=}2} \rightarrow \langle\langle s, c\rangle\rangle\bigcirc(\mathsf{x{=}0} \wedge \neg E_{s,c}\mathsf{x{=}0})$: both players can cooperate to enforce q_0 as the next state, but they cannot impose on themselves the epistemic ability to recognize the state;

5. $\mathsf{x{=}0} \rightarrow D_{s,c}\mathsf{x{=}0}$. In q_0, the server can rule out the possibility that q_1 is the case, and the client can rule out q_2. Thus, they know that the current state is exactly q_0 if they are allowed to combine their knowledge.

□

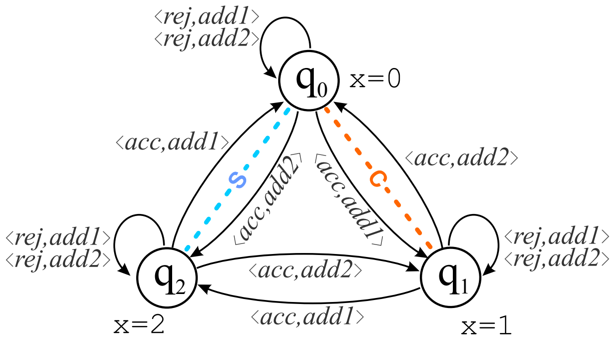


Figure 3.3: A multi-player epistemic game model for the modified controller/client problem

3.3.2 Extending Multi-Player Game Models and Coalition Effectivity Models to Include Knowledge

Multi-player game models and coalition effectivity models can be augmented with epistemic accessibility relations in a similar way, giving way to multi-player epistemic game models $\mathcal{M} = \langle Q, \gamma, \pi, \sim_{a_1}, ..., \sim_{a_k} \rangle$ and epistemic coalition effectivity models $\mathcal{E} = \langle \mathbb{A}\mathrm{gt}, Q, E, \pi, \sim_{a_1}, ..., \sim_{a_k} \rangle$ for a set of agents $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$ over a set of propositions Π. Semantic rules for epistemic formulae remain the same as in Section 3.3.1 for both kinds of structures. The equivalence results from Section 2.7 can be extended to ATEL and its models.

Corollary 3.3 *For every* ATEL *formula* φ *the following are equivalent:*

1. φ *is valid in all alternating epistemic transition systems,*
2. φ *is valid in all multi-player epistemic game models,*
3. φ *is valid in all standard epistemic coalition effectivity models.*

Example 3.3 A multi-player epistemic game model for the modified controller/client system from Example 3.2 is shown in Figure 3.3. The same formulae are valid. □

We will use multi-player epistemic game models throughout the rest of this chapter for the convenience of presentation they offer.

3.3.3 Problems with ATEL

One of the main challenges in ATEL is the question how, given an explicit way to represent agents' knowledge, this should interfere with the agents' available strategies. What does it mean that an agent has a strategy to enforce φ, if it involves making different choices in states that are epistemically indistinguishable for the agent, for instance? Moreover, agents are assumed some epistemic capabilities when making decisions, and other for epistemic properties like $K_a\varphi$. The interpretation of knowledge operators refers to the agents' capability to distinguish one *state* from another; the semantics of $\langle\langle A \rangle\rangle$ allows the agents to base their decisions upon *sequences* of states. These relations between complete vs. incomplete information on one hand, and perfect vs. imperfect recall on the other, are studied in Chapter 4 in more detail. It is also argued that, when reasoning about what an agent can *enforce*, it seems more appropriate to require the agent to know his winning strategy rather than to know only that such a strategy exists. Two variations of ATEL are proposed as solutions: Alternating-time Temporal Observational Logic ATOL (Section 4.4) for agents with bounded memory and syntax restricted in a way similar to CTL, and full Alternating-time Temporal Epistemic Logic with Recall ATEL-R* (Section 4.5), where agents are able to memorize the whole game. We believe that analogous results to those presented here about ATEL can be obtained for logics like ATOL and ATEL-R* and their models.

3.4 Interpretations of ATEL into ATL

ATL is trivially embedded into ATEL since all ATL formulae are also ATEL formulae. Moreover, every multi-player game model can be extended to a multi-player epistemic game model by defining all epistemic accessibility relations to be the equality, i.e. all agents have no uncertainty about the current state of the system – thus embedding the semantics of ATL in the one for ATEL, and rendering the former a reduct of the latter.

Finding an interpretation the other way is more involved. We will first construct a satisfiability preserving interpretation of the fragment of ATEL without distributed knowledge (we will call it ATEL$_{CE}$), and then we will show how it can be extended to the whole ATEL, though at the expense of some blow-up of the models. The interpretation we propose has been inspired by (Schild, 2000). We should also mention (van Otterloo et al., 2003), as it deals with virtually the same issue. Related work is discussed in more detail at the end of the section.

3.4.1 Idea of the Interpretation

ATEL consists of two orthogonal layers. The first one, inherited from ATL, refers to what agents can achieve in temporal perspective, and is underpinned by the structure defined via transition function o. The other layer is the epistemic component, reflected by epistemic accessibility relations. Our idea of the translation is to leave the original temporal structure intact, while extending it with additional transitions to "simulate" epistemic accessibility links. The "simulation" – like the one in (van Otterloo et al., 2003) – is achieved through adding new "epistemic" agents, who can enforce transitions to epistemically accessible states. Unlike in that paper, though, the "moves" of epistemic agents are orthogonal to the original temporal transitions ("action" transitions): they lead to special "epistemic" copies of the original states rather than to the "action" states themselves, and no new states are introduced into the course of action. The "action" and "epistemic" states form separate strata in the resulting model, and are labeled accordingly to distinguish transitions that implement different modalities.

The interpretation consists of two independent parts: a transformation of models and a translation of formulae. First, we propose a construction that transforms every

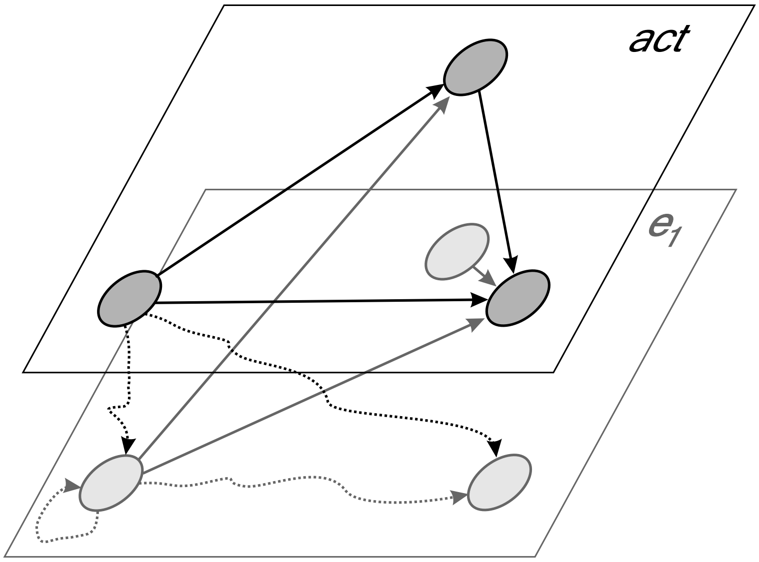


Figure 3.4: New model: "action" vs. "epistemic" states, and "action" vs. "epistemic" transitions. Note that the game frames for "epistemic" states are *exact* copies of their "action" originals: the "action" transitions from the epistemic layer lead back to the "action" states.

multi-player epistemic game model $\mathcal{M}$ for a set of agents $\{a_1, ..., a_k\}$, into a (pure) multi-player game model $\mathcal{M}^{\text{ATL}}$ over a set of agents $\{a_1, ..., a_k, e_1, ..., e_k\}$. Agents $a_1, ..., a_k$ are the original agents from M (we will call them "real agents"). Agents $e_1, ..., e_k$ are "epistemic doubles" of the real agents: the role of e_i is to "point out" the states that were epistemically indistinguishable from the current state for agent a_1 in $\mathcal{M}$. Intuitively, $K_{a_i}\varphi$ could be then replaced with a formula like $\neg\langle\langle e_i\rangle\rangle\bigcirc\neg\varphi$ that rephrases the semantic definition of K_a operator from Section 3.3.1. As $\mathcal{M}^{\text{ATL}}$ inherits the temporal structure from $\mathcal{M}$, temporal formulae might be left intact. However, it is not as simple as that.

Note that agents make their choices simultaneously in multi-player game models, and the resulting transition is a result of all these choices. In consequence, it is not possible that an epistemic agent e_i can enforce an "epistemic" transition to state q, and at the same time a group of real agents A is capable of executing an "action" transition to q'. Thus, in order to distinguish transitions referring to different modalities, we introduce additional states in model $\mathcal{M}^{\text{ATL}}$. States $q_1^{\mathsf{e_i}}, ..., q_n^{\mathsf{e_i}}$ are exact copies of the original states $q_1, ..., q_n$ from Q except for one thing: they satisfy a new proposition $\mathsf{e_i}$, added to enable identifying moves of epistemic agent e_i. Original states $q_1, ..., q_n$ are still in $\mathcal{M}^{\text{ATL}}$ to represent targets of "action" moves of the real agents $a_1, ..., a_k$. We will use a new proposition act to label these states. The type of a transition can be recognized by the label of its target state (cf. Figure 3.4).

Now, we must only arrange the interplay between agents' choices, so that the results can be interpreted in a direct way. To achieve this, every epistemic agent can choose to be "passive" and let the others decide upon the next move, or may select one of the states indistinguishable from q for an agent a_i (to be more precise, the epistemic agents do select the epistemic copies of states from $Q^{\mathsf{e_i}}$ rather than the original action states from Q). The resulting transition leads to the state selected by the *first* non-passive epistemic agent. If all the epistemic agents decided to be passive, the "action" transition chosen by the real agents follows.

For such a construction of $\mathcal{M}^{\mathrm{ATL}}$, we can finally show how to translate formulae from ATEL to ATL:

- $K_{a_i}\varphi$ can be rephrased as $\neg\langle\langle\{e_1, ..., e_i\}\rangle\rangle\bigcirc(\mathsf{e_i} \wedge \neg\varphi)$: the epistemic moves to agent e_i's epistemic states do not lead to a state where φ fails. Note that player e_i can select a state of his if, and only if, players $e_1, ..., e_{i-1}$ are passive (hence their presence in the cooperation modality). Note also that $K_{a_i}\varphi$ can be as well translated as $\neg\langle\langle\{e_1, ..., e_k\}\rangle\rangle\bigcirc(\mathsf{e_i} \wedge \neg\varphi)$ or $\neg\langle\langle\{a_1, ..., a_k, e_1, ..., e_k\}\rangle\rangle\bigcirc(\mathsf{e_i} \wedge \neg\varphi)$: when e_i decides to be active, choices from $a_1, ..., a_k$ and $e_{i+1}, ..., e_k$ are irrelevant.

- $\langle\langle A\rangle\rangle\bigcirc\varphi$ becomes $\langle\langle A \cup \{e_1, ..., e_k\}\rangle\rangle\bigcirc(\mathsf{act} \wedge \varphi)$ in a similar way.

- To translate other temporal formulae, we must require that the relevant part of a path runs only through "action" states (labeled with act proposition). Thus, $\langle\langle A\rangle\rangle\Box\varphi$ can be rephrased as $\varphi \wedge \langle\langle A \cup \mathbb{A}\mathrm{gt}^e\rangle\rangle\bigcirc\langle\langle A \cup \mathbb{A}\mathrm{gt}^e\rangle\rangle\Box(\mathsf{act} \wedge \varphi)$. Note that a simpler translation with $\langle\langle A \cup \mathbb{A}\mathrm{gt}^e\rangle\rangle\Box(\mathsf{act} \wedge \varphi)$ is incorrect: the initial state of a path does not have to be an action state, since $\langle\langle A\rangle\rangle\Box\varphi$ can be embedded in an epistemic formula. A similar method applies to the translation of $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$.

- Translation of common knowledge refers to the definition of relation $\sim_A^C$ as the transitive closure of relations $\sim_{a_i}$: $C_A\varphi$ means that all the (finite) sequences of appropriate epistemic transitions must end up in a state where φ is true.

The only operator that does not seem to lend itself to a translation according to the above scheme is the distributed knowledge operator D, for which we seem to need more "auxiliary" agents. Thus, we will begin with presenting details of our interpretation for ATEL$_{CE}$ – a reduced version of ATEL that includes only common knowledge and "everybody knows" operators for group epistemics. Section 3.4.4 shows how to modify the translation to include distributed knowledge as well.

Since the interpretation yields a polynomial model checking algorithm for ATEL, it can be used for multi-agent planning that involves epistemic goals, using the "planning as model checking" idea discussed already in Section 2.8. A few examples of such "planning for epistemic goals" are shown in Section 3.4.5. One must be cautious, however, while reading $\langle\langle A\rangle\rangle\Phi$ as "team A has a *plan* to enforce Φ" in the context of ATEL and the incomplete information assumption. ATEL semantics implies unexpected properties for many models, which is especially evident when we understand "having a strategy" as "being in control to execute a plan". We investigate the issue extensively in Chapter 4.

The interpretation we propose in Sections 3.4.3 and 3.4.4 can be extended to handle the more general language of ATEL*, as shown in Section 3.4.6. Finally, we can also modify the interpretation to show how to translate formulae of the propositional version of BDI logic from (Schild, 2000) into ATL (and even to CTL) in Section 3.5.1.

3.4.2 Related Work

The interpretation presented in this section has been inspired by (Schild, 2000) in which a propositional variant of the BDI logic (Rao and Georgeff, 1991) was proved to be subsumed by propositional μ-calculus. We use a similar method here to show a translation from ATEL models and formulae to models and formulae of ATL that preserves satisfiability. ATL (just like μ-calculus) is a multi-modal logic, where modalities are indexed by agents (programs in the case of μ-calculus). It is therefore possible to "simulate" the epistemic layer of ATEL by adding new agents (and hence new cooperation modalities) to the scope. Thus, the general idea of the interpretation is to translate modalities of one kind to additional modalities of another kind.

Similar translations are well known within modal logics community, including translation of epistemic logic into Propositional Dynamic Logic, translation of dynamic epistemic logic without common knowledge into epistemic logic (Gerbrandy, 1999) etc. A work particularly close to ours is included in (van Otterloo et al., 2003). In that paper, a reduction of ATEL model checking to model checking of ATL formulae is presented, and the epistemic accessibility relations are handled in a similar way to our approach, i.e. with use of additional "epistemic" agents. We believe, however, that our translation is more general, and provides more flexible framework in many respects:

1. The algorithm from (van Otterloo et al., 2003) is intended only for *turn-based acyclic* transition systems, which is an essential limitation of its applicability. Moreover, the set of states is assumed to be finite (hence only finite trees are considered). There is no restriction like this in our method.

2. The language of ATL/ATEL is distinctly reduced in (van Otterloo et al., 2003): it includes only "sometime" ($\Diamond$) and "always" ($\Box$) operators in the temporal part (neither "next" nor "until" are treated), and the individual knowledge operator K_a (the group knowledge operators C, E, D are absent).

3. The translation of a model in (van Otterloo et al., 2003) depends heavily on the formula one wants to model-check, while in the algorithm presented here, formulae and models are translated independently (except for the sole case of efficient translation of distributed knowledge).

4. Our intuition is that our interpretation is also more general in the sense that it can work in contexts other than model checking. We plan to apply the same translation scheme to reduce the logic of ATEL itself, i.e. to reduce ATEL to ATL. Given decidability and a complete axiomatization of ATL (Goranko and van Drimmelen, 2003), such a reduction would carry the results over to ATEL.

3.4.3 Interpreting Models and Formulae of ATEL$_{CE}$ into ATL

Given a multi-player epistemic game model $\mathcal{M} = \langle Q, \gamma, \pi, \sim_{a_1}, ..., \sim a_k \rangle$ for a set of agents $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$ over a set of propositions Π, we construct a new game model $\mathcal{M}^{\mathrm{ATL}} = \langle Q', \gamma', \pi' \rangle$ over a set of agents $\mathbb{A}\mathrm{gt}' = \mathbb{A}\mathrm{gt} \cup \mathbb{A}\mathrm{gt}^e$, where:

- $\mathbb{A}\mathrm{gt}^e = \{e_1, ..., e_k\}$ is the set of epistemic agents;
- $Q' = Q \cup Q^{\mathsf{e}_1} \cup ... \cup Q^{\mathsf{e}_k}$, where $Q^{\mathsf{e}_i} = \{q^{\mathsf{e}_i} \mid q \in Q\}$. We assume that $Q, Q^{\mathsf{e}_1}, ..., Q^{\mathsf{e}_k}$ are pairwise disjoint. Further we will be using the more general notation $S^{\mathsf{e}_i} = \{q^{\mathsf{e}_i} \mid q \in S\}$ for any $S \subseteq Q$.
- $\Pi' = \Pi \cup \{\mathsf{act}, \mathsf{e}_1, ..., \mathsf{e}_k\}$, and $\pi'(p) = \pi(p) \cup \bigcup_{i=1,...,k} \pi(p)^{\mathsf{e}_i}$ for every proposition $p \in \Pi$. Moreover, $\pi'(\mathsf{act}) = Q$ and $\pi'(\mathsf{e_i}) = Q^{\mathsf{e}_i}$.

We assume that all the epistemic agents from $\mathbb{A}\mathrm{gt}^e$, states from $States^{\mathsf{e}_1} \cup ... \cup Q^{\mathsf{e}_k}$, and propositions from $\{\mathsf{act}, \mathsf{e}_1, ..., \mathsf{e}_k\}$, are *new* and have been absent in the original model $\mathcal{M}$.

For every state q in $\mathcal{M}$, we translate the frame $\gamma(q) = \langle \mathbb{A}\mathrm{gt}, \{\Sigma_a^q \mid a \in \mathbb{A}\mathrm{gt}\}, Q, o \rangle$ to $\gamma'(q) = \langle \mathbb{A}\mathrm{gt}', \{\Sigma_a^{q'} \mid a \in \mathbb{A}\mathrm{gt}'\}, Q', o' \rangle$:

- $\Sigma_a^{q'} = \Sigma_a^q$ for $a \in \mathbb{A}\mathrm{gt}$: choices of the "real" agents do not change;
- $\Sigma_{e_i}^{q'} = \{pass\} \cup img(q, \sim_{a_i})^{\mathsf{e}_i}$ for $i = 1, ..., k$, where $img(q, R) = \{q' \mid qRq'\}$ is the image of q with respect to relation R.
- the new transition function is defined as follows:

$$o'_q(\sigma_{a_1}, ..., \sigma_{a_k}, \sigma_{e_1}, ..., \sigma_{e_k}) = \begin{cases} o_q(\sigma_{a_1}, ..., \sigma_{a_k}) & \text{if } \sigma_{e_1} = ... = \sigma_{e_k} = pass \\ \sigma_{e_i} & \text{if } e_i \text{ is the first active epistemic agent.} \end{cases}$$

The game frames for the new states are exactly the same: $\gamma'(q^{\mathsf{e}_i}) = \gamma'(q)$ for all $i = 1, ..., k, q \in Q$.

Example 3.4 A part of the resulting structure for the epistemic game model from Figure 3.3 is shown in Figure 3.5. All the new states, plus the transitions going out of q_2 are presented. The wildcard "*" stands for any action of the respective agent. For instance, $\langle reject, *, pass, pass \rangle$ represents $\langle reject, set0, pass, pass \rangle$ and $\langle reject, set1, pass, pass \rangle$. □

Now, we define a translation of formulae from ATEL$_{CE}$ to ATL corresponding to the above described interpretation of ATEL models into ATL models:

$$\begin{aligned} tr(p) &= p, \qquad \text{for } p \in \Pi \\ tr(\neg\varphi) &= \neg tr(\varphi) \\ tr(\varphi \vee \psi) &= tr(\varphi) \vee tr(\psi) \\ tr(\langle\langle A \rangle\rangle \bigcirc \varphi) &= \langle\langle A \cup \mathbb{A}\mathrm{gt}^e \rangle\rangle \bigcirc (\mathsf{act} \wedge tr(\varphi)) \end{aligned}$$

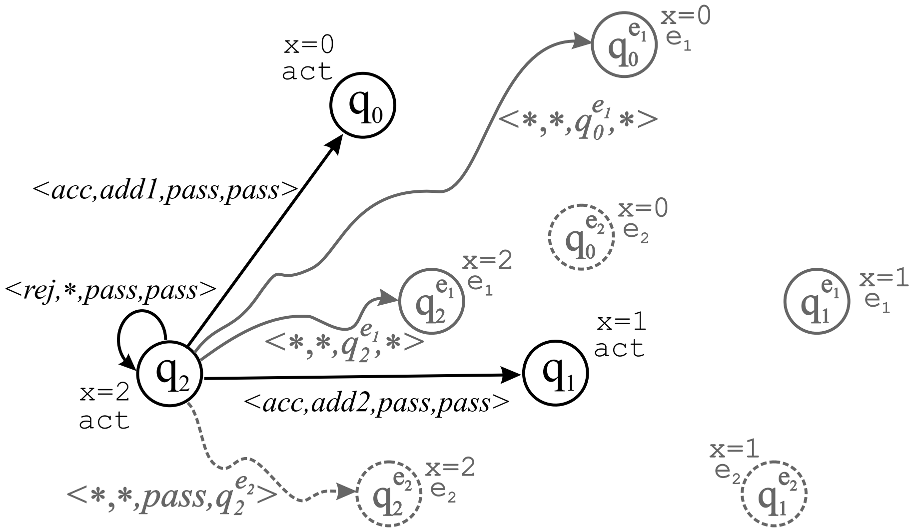


Figure 3.5: Construction for the multi-player epistemic game model from Figure 3.3.

$$
\begin{aligned}
tr(\langle\!\langle A\rangle\!\rangle\Box\varphi) &= tr(\varphi)\wedge\langle\!\langle A\cup\mathbb{A}\mathrm{gt}^e\rangle\!\rangle\bigcirc\langle\!\langle A\cup\mathbb{A}\mathrm{gt}^e\rangle\!\rangle\Box(\mathsf{act}\wedge tr(\varphi))\\
tr(\langle\!\langle A\rangle\!\rangle\varphi\mathcal{U}\psi) &= tr(\psi)\vee(tr(\varphi)\wedge\langle\!\langle A\cup\mathbb{A}\mathrm{gt}^e\rangle\!\rangle\bigcirc\langle\!\langle A\cup\mathbb{A}\mathrm{gt}^e\rangle\!\rangle\\
&\qquad(\mathsf{act}\wedge tr(\varphi))\,\mathcal{U}(\mathsf{act}\wedge tr(\psi)))\\
tr(K_{a_i}\varphi) &= \neg\langle\!\langle\{e_1,...,e_i\}\rangle\!\rangle\bigcirc(\mathsf{e_i}\wedge\neg tr(\varphi))\\
tr(E_A\varphi) &= \neg\langle\!\langle\mathbb{A}\mathrm{gt}^e\rangle\!\rangle\bigcirc(\bigvee_{a_i\in A}\mathsf{e_i}\wedge\neg tr(\varphi))\\
tr(C_A\varphi) &= \neg\langle\!\langle\mathbb{A}\mathrm{gt}^e\rangle\!\rangle\bigcirc\langle\!\langle\mathbb{A}\mathrm{gt}^e\rangle\!\rangle(\bigvee_{a_i\in A}\mathsf{e_i})\,\mathcal{U}(\bigvee_{a_i\in A}\mathsf{e_i}\wedge\neg tr(\varphi))
\end{aligned}
$$

Lemma 3.4 *For every* ATEL$_{CE}$ *formula* φ*, model* $\mathcal{M}$*, and "action" state* $q\in Q$*, we have* $\mathcal{M}^{\mathrm{ATL}}, q\models tr(\varphi)$ *iff* $\mathcal{M}^{\mathrm{ATL}}, q^{\mathsf{e_i}}\models tr(\varphi)$ *for every* $i=1,...,k$*.*

Proof sketch (structural induction on φ)**:** It suffices to note that $tr(\varphi)$ cannot contain propositions $\mathsf{act}, \mathsf{e_1}, ..., \mathsf{e_k}$ outside of the scope of $\langle\!\langle A\rangle\!\rangle\bigcirc$ for some $A\subseteq\mathbb{A}\mathrm{gt}'$. Besides, the propositions from φ are true in q iff they are true in $q^{\mathsf{e_1}}, ..., q^{\mathsf{e_k}}$ and the game frames for $q, q^{\mathsf{e_1}}, ..., q^{\mathsf{e_k}}$ are the same. □

Lemma 3.5 *For every* ATEL$_{CE}$ *formula* φ*, model* $\mathcal{M}$*, and "action" state* $q\in Q$*, we have* $\mathcal{M}, q\models\varphi$ *iff* $\mathcal{M}^{\mathrm{ATL}}, q\models tr(\varphi)$*.*

Proof: The proof follows by structural induction on φ. We will show that the construction preserves the truth value of φ for three cases: $\varphi\equiv\langle\!\langle A\rangle\!\rangle\bigcirc\psi$, $\varphi\equiv\langle\!\langle A\rangle\!\rangle\Box\psi$ and $\varphi\equiv C_A\psi$. An interested reader can tackle the other cases in an analogous way.

case $\varphi\equiv\langle\!\langle A\rangle\!\rangle\bigcirc\psi$**, ATEL$_{CE}$ $\Rightarrow$ ATL.** Let $\mathcal{M}, q\models\langle\!\langle A\rangle\!\rangle\bigcirc\psi$, then there is σ_A such that for every $\sigma_{\mathbb{A}\mathrm{gt}\setminus A}$ we have $o_q(\sigma_A,\sigma_{\mathbb{A}\mathrm{gt}\setminus A})\models\psi$. By induction hypothesis,

$\mathcal{M}^{\text{ATL}}, o_q(\sigma_A, \sigma_{\mathbb{A}\text{gt}\setminus A}) \models tr(\psi)$; also, $\mathcal{M}^{\text{ATL}}, o_q(\sigma_A, \sigma_{\mathbb{A}\text{gt}\setminus A}) \models \mathsf{act}$. Thus, $\mathcal{M}^{\text{ATL}}, o'_q(\sigma_A, \sigma_{\mathbb{A}\text{gt}\setminus A}, pass_{\mathbb{A}\text{gt}^e}) = o_q(\sigma_A, \sigma_{\mathbb{A}\text{gt}\setminus A}) \models \mathsf{act} \wedge tr(\psi)$, where $pass_C$ denotes the strategy where every agent from $C \subseteq \mathbb{A}\text{gt}^e$ decides to be passive. In consequence, $\mathcal{M}^{\text{ATL}}, q \models \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \bigcirc tr(\psi)$.

case $\varphi \equiv \langle\langle A\rangle\rangle \bigcirc \psi$**, ATL ⇒ ATEL**$_{CE}$**.** $\mathcal{M}^{\text{ATL}}, q \models \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \bigcirc(\mathsf{act} \wedge tr(\psi))$, so there is $\sigma_{A\cup\mathbb{A}\text{gt}^e}$ such that for every $\sigma_{\mathbb{A}\text{gt}'\setminus(A\cup\mathbb{A}\text{gt}^e)} = \sigma_{\mathbb{A}\text{gt}\setminus A}$ we have that $\mathcal{M}^{\text{ATL}}, o'_q(\sigma_{A\cup\mathbb{A}\text{gt}^e}, \sigma_{\mathbb{A}\text{gt}\setminus A}) \models \mathsf{act} \wedge tr(\psi)$. Note also that act is true in $\mathcal{M}^{\text{ATL}}$ and $o'_q(\sigma_{A\cup\mathbb{A}\text{gt}^e}, \sigma_{\mathbb{A}\text{gt}\setminus A})$ only when $\sigma_{A\cup\mathbb{A}\text{gt}^e} = \langle\sigma_A, pass_{\mathbb{A}\text{gt}^e}\rangle$, otherwise the transition would lead to an epistemic state. In consequence, we have that $o'_q(\sigma_{A\cup\mathbb{A}\text{gt}^e}, \sigma_{\mathbb{A}\text{gt}\setminus A}) = o_q(\sigma_A, \sigma_{\mathbb{A}\text{gt}\setminus A})$, and hence $\mathcal{M}^{\text{ATL}}, o_q(\sigma_A, \sigma_{\mathbb{A}\text{gt}\setminus A}) \models tr(\psi)$. By the induction hypothesis, $\mathcal{M}, o_q(\sigma_A, \sigma_{\mathbb{A}\text{gt}\setminus A}) \models \psi$ and finally $\mathcal{M}, q \models \langle\langle A\rangle\rangle \bigcirc \psi$.

case $\varphi \equiv \langle\langle A\rangle\rangle \Box \psi$**, ATEL**$_{CE}$ **⇒ ATL.** Let $\mathcal{M}, q \models \langle\langle A\rangle\rangle \Box \psi$, then A have a collective strategy F_A such that for every $\Lambda \in out_{\mathcal{M}}(q, F_A)$ and $i \geq 0$ we have $\mathcal{M}, \Lambda[i] \models \psi$ (*). Consider a strategy $F'_{A\cup\mathbb{A}\text{gt}^e}$ in the new model $\mathcal{M}^{\text{ATL}}$, such that $F'_{A\cup\mathbb{A}\text{gt}^e}(a)(\lambda) = F_A(a)(\lambda)$ for all $a \in A$ and $\lambda \in Q^+$, and $F'_{A\cup\mathbb{A}\text{gt}^e}(a)(\lambda) = pass$ for all $a \in \mathbb{A}\text{gt}^e$. In other words, $F'_{A\cup\mathbb{A}\text{gt}^e}$ is a strategy according to which the agents from A do exactly the same as in the original strategy F_A for all the histories of "action" states from Q (and do anything for all the other histories), while the epistemic agents remain passive all the time. Since all the epistemic agents pass, every $\Lambda \in out_{\mathcal{M}^{\text{ATL}}}(q, F'_{A\cup\mathbb{A}\text{gt}^e})$ includes only "action" states from Q (**). As the agents from A make the same choices as in F_A, we obtain that $out_{\mathcal{M}^{\text{ATL}}}(q, F'_{A\cup\mathbb{A}\text{gt}^e}) = out_{\mathcal{M}}(q, F_A)$. By (*) and the induction hypothesis: for every $\Lambda \in out_{\mathcal{M}^{\text{ATL}}}(q, F'_{A\cup\mathbb{A}\text{gt}^e})$ and $i \geq 0$ we have $\mathcal{M}, \Lambda[i] \models tr(\psi)$. By (**), also $\mathcal{M}^{\text{ATL}}, \Lambda[i] \models \mathsf{act}$ for all such Λ and i. In consequence, $\mathcal{M}^{\text{ATL}}, q \models \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \Box(tr(\psi) \wedge \mathsf{act})$, which implies that $\mathcal{M}^{\text{ATL}}, q \models tr(\psi) \wedge \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \bigcirc \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \Box(tr(\psi) \wedge \mathsf{act})$.[2]

case $\varphi \equiv \langle\langle A\rangle\rangle \Box \psi$**, ATL ⇒ ATEL**$_{CE}$**.** Let $\mathcal{M}^{\text{ATL}}$ and an "action" state $q \in Q$ satisfy $tr(\psi) \wedge \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \bigcirc \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \Box(tr(\psi) \wedge \mathsf{act})$. Note that $\mathcal{M}^{\text{ATL}}, q \models \mathsf{act}$, so $\mathcal{M}^{\text{ATL}}, q \models (tr(\psi) \wedge \mathsf{act}) \wedge \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \bigcirc \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \Box(tr(\psi) \wedge \mathsf{act})$, which is equivalent to $\mathcal{M}^{\text{ATL}}, q \models \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \Box(tr(\psi) \wedge \mathsf{act})$. Thus, $A \cup \mathbb{A}\text{gt}^e$ have a collective strategy $F_{A\cup\mathbb{A}\text{gt}^e}$ such that for every $\Lambda \in out_{\mathcal{M}^{\text{ATL}}}(q, F_{A\cup\mathbb{A}\text{gt}^e})$ and $i \geq 0$ we have $\mathcal{M}^{\text{ATL}}, \Lambda[i] \models \mathsf{act} \wedge tr(\psi)$. In consequence, $\mathcal{M}^{\text{ATL}}, \Lambda[i] \models \mathsf{act}$ (*) and $\mathcal{M}^{\text{ATL}}, \Lambda[i] \models tr(\psi)$ (**). By (*), Λ includes only "action" states from Q, and hence Λ is a path in $\mathcal{M}$ as well. Moreover, (*) implies also that $F_{A\cup\mathbb{A}\text{gt}^e}(a)(\lambda) = pass$ for every $a \in \mathbb{A}\text{gt}^e$ and λ being any finite prefix of

[2]The proof suggests a simpler translation of $\langle\langle A\rangle\rangle \Box \psi$: namely, $\langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \Box(tr(\psi) \wedge \mathsf{act})$ instead of $tr(\psi) \wedge \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \bigcirc \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle \Box(tr(\psi) \wedge \mathsf{act})$ (cf. also Remark 3.9). Indeed, no part of the proof of Lemma 3.5 *directly* rules out the simpler scheme. Note, however, that the proof of Lemma 3.5 uses the result from Lemma 3.4 in several places, and the proof of Lemma 3.5 *does* depend on the fact that proposition act occurs only in the scope of $\langle\langle \Gamma\rangle\rangle \bigcirc$.

a computation from $out_{\mathcal{M}^{\text{ATL}}}(q, F_{A\cup\mathbb{A}\text{gt}^e})$. Thus, the actual sequence of "action" states in computation Λ entirely depends on choices of the "real" agents from $\mathbb{A}\text{gt}$.

Let F'_A, such that $F'_A(a)(\lambda) = F_{A\cup\mathbb{A}\text{gt}^e}(a)(\lambda)$ for every $a \in A$ and $\lambda \in Q^+$, be a collective strategy for A in $\mathcal{M}$. Then, $out_{\mathcal{M}}(q, F'_A) = out_{\mathcal{M}^{\text{ATL}}}(q, F_{A\cup\mathbb{A}\text{gt}^e})$. By (**) and the induction hypothesis, $\mathcal{M}, \Lambda[i] \models \psi$ for every $\Lambda \in out_{\mathcal{M}}(q, F_A)$ and $i \geq 0$. In consequence, $\mathcal{M}, q \models \langle\langle A\rangle\rangle\Box\psi$.

case $\varphi \equiv C_A\psi$, ATEL$_{CE}$ $\Rightarrow$ ATL. We have $\mathcal{M}, q \models C_A\psi$, so for every sequence of states $q_0 = q, q_1, ..., q_n$, $q_i \sim_{a_{j_i}} q_{i+1}$, $a_{j_i} \in A$ for $i = 0, ..., n-1$, it is true that $\mathcal{M}, q_n \models \psi$. Consider the same q in $\mathcal{M}^{\text{ATL}}$. The shape of the construction implies that for every sequence $q'_0 = q, q'_1, ..., q'_n$ in which every q_{i+1} is a successor of q_i and every $q_{i+1} \in Q^{\mathsf{e_{j_i}}}$, $e_{j_i} \in A^e$, we have $\mathcal{M}^{\text{ATL}}, q'_n \models tr(\psi)$ (by induction and Lemma 3.4). Moreover, $\mathcal{M}^{\text{ATL}}, q'_i \models \mathsf{e_{j_i}}$ for $i \geq 1$, hence $\mathcal{M}^{\text{ATL}}, q'_i \models \bigvee_{a_j\in A} \mathsf{e_j}$. Note that the above refers to all the sequences that can be enforced by the agents from $\mathbb{A}\text{gt}^e$, and have $\bigvee_{a_j\in A} \mathsf{e_j}$ true along the way (from q'_1 on). Thus, $\mathbb{A}\text{gt}^e$ have no strategy from q such that $\bigvee_{a_j\in A} \mathsf{e_j}$ holds from the next state on, and $tr(\psi)$ is eventually false:
$\mathcal{M}^{\text{ATL}}, q \not\models_{\text{ATL}} \langle\langle\mathbb{A}\text{gt}^e\rangle\rangle\bigcirc\langle\langle\mathbb{A}\text{gt}^e\rangle\rangle(\bigvee_{a_j\in A}\mathsf{e_j})\,\mathcal{U}(\bigvee_{a_j\in A}\mathsf{e_j} \wedge \neg tr(\psi))$,
which proves the case.

case $\varphi \equiv C_A\psi$, ATL $\Rightarrow$ ATEL$_{CE}$. We have
$\mathcal{M}^{\text{ATL}}, q \models \neg\langle\langle\mathbb{A}\text{gt}^e\rangle\rangle\bigcirc\langle\langle\mathbb{A}\text{gt}^e\rangle\rangle(\bigvee_{a_j\in A}\mathsf{e_j})\,\mathcal{U}(\bigvee_{a_j\in A}\mathsf{e_j} \wedge \neg tr(\psi))$, so for every $\sigma_{\mathbb{A}\text{gt}^e}$ there is $\sigma_{\mathbb{A}\text{gt}'\setminus\mathbb{A}\text{gt}^e} = \sigma_{\mathbb{A}\text{gt}}$ such that $o'_q(\sigma_{\mathbb{A}\text{gt}^e}, \sigma_{\mathbb{A}\text{gt}}) = q' \in Q'$ and $\mathcal{M}^{\text{ATL}}, q' \models \neg\langle\langle\mathbb{A}\text{gt}^e\rangle\rangle(\bigvee_{a_j\in A}\mathsf{e_j})\,\mathcal{U}(\bigvee_{a_j\in A}\mathsf{e_j} \wedge \neg tr(\psi))$. In particular, this implies that the above holds for all epistemic states q' that are successors of q in $\mathcal{M}^{\text{ATL}}$, also the ones that refer to agents from A (*).

Suppose that $\mathcal{M}, q \not\models C_A\psi$ (**). Let us now take the action counterpart $q'_{\mathsf{act}} \in Q$ of q'. By (*), (**) and properties of the construction, q'_{act} occurs also in $\mathcal{M}$, and there must be a path $q_0 = q, q_1 = q'_{\mathsf{act}}, ..., q_n$, $q_i \in Q$, such that $q_i \sim_{a_{j_i}} q_{i+1}$ and $\mathcal{M}, q_n \not\models_{ATEL} \psi$. Then, $\mathcal{M}^{\text{ATL}}, q_n \not\models_{\text{ATL}} tr(\psi)$ (by induction). This means also that we have a sequence $q_0 = q, q'_1 = q', ..., q'_n$ in $\mathcal{M}^{\text{ATL}}$, in which every $q'_i \in Q^{\mathsf{e_{j_i}}}$, $a_{j_i} \in A$, is an epistemic counterpart of q_i. Thus, for every $i = 1, ..., n$: $\mathcal{M}^{\text{ATL}}, q'_i \models \mathsf{e_{j_i}}$, so $\mathcal{M}^{\text{ATL}}, q'_i \models \bigvee_{a_j\in A}\mathsf{e_j}$. Moreover, $\mathcal{M}^{\text{ATL}}, q_n \not\models_{\text{ATL}} tr(\psi)$ implies that $\mathcal{M}^{\text{ATL}}, q'_n \not\models_{\text{ATL}} tr(\psi)$ (by Lemma 3.4), so $\mathcal{M}^{\text{ATL}}, q'_n \models \neg tr(\psi)$. Thus, $\mathcal{M}^{\text{ATL}}, q' \models \langle\langle\mathbb{A}\text{gt}^e\rangle\rangle(\bigvee_{a_j\in A}\mathsf{e_j})\,\mathcal{U}(\bigvee_{a_j\in A}\mathsf{e_j} \wedge \neg tr(\psi))$, which contradicts (*).

□

As an immediate corollary of the last two lemmata we obtain:

Theorem 3.6 *For every* ATEL$_{CE}$ *formula φ and model $\mathcal{M}$, φ is satisfiable (resp. valid) in $\mathcal{M}$ iff $tr(\varphi)$ is satisfiable (resp. valid) in $\mathcal{M}^{\text{ATL}}$.*

Note that the construction used above to interpret ATEL$_{CE}$ in ATL has several nice complexity properties:

- The vocabulary (set of propositions Π) only increases linearly (and certainly remains finite).
- The set of states in an ATEL-model grows linearly, too: if model $\mathcal{M}$ includes n states and k agents, then $\mathcal{M}^{\text{ATL}}$ has $n' = (k+1)n = O(kn)$ states.
- Let m be the number of transitions in $\mathcal{M}$. We have $(k+1)m$ action transitions in $\mathcal{M}^{\text{ATL}}$. Since the size of every set $img(q, \sim_a)$ can be at most n, there may be no more than kn epistemic transitions per state in $\mathcal{M}^{\text{ATL}}$, hence at most $(k+1)nkn$ in total. Because $m \leq n^2$, we have $m' = O(k^2 n^2)$.
- Only the length of formulae may suffer an exponential blow-up, because the translation of $\langle\langle A\rangle\rangle\Box\varphi$ involves two occurrences of $tr(\varphi)$, and the translation of $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$ involves two occurrences of both $tr(\varphi)$ and $tr(\psi)$. Thus, every nesting of $\langle\langle A\rangle\rangle\Box\varphi$ and $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$ roughly doubles the size of the translated formula in the technical sense. However, the number of *different subformulae* in the formula only increases linearly. Note that the automata-based methods for model checking (Alur et al., 2002) or satisfiability checking (van Drimmelen, 2003) for ATL are based on an automaton associated with the given formula, built from its "subformulae closure" – and their complexity depends on the number of different subformulae in the formula rather than number of symbols.

  In fact, we can avoid the exponential growth of formulae in the context of satisfiability checking by introducing a new propositional variable p and requiring that it is universally equivalent to $tr(\varphi)$, i.e. adding conjunct $\langle\langle\varnothing\rangle\rangle\Box(p \leftrightarrow tr(\varphi))$ to the whole translated formula. Then $\langle\langle A\rangle\rangle\Box\varphi$ can be simply translated as $p \wedge \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle\bigcirc\langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle\Box(\mathsf{act} \wedge p)$. "Until" formulae $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$ are treated analogously. A similar method can be proposed for model checking. To translate $\langle\langle A\rangle\rangle\Box\varphi$, we first use the algorithm from (Alur et al., 2002) and model-check $tr(\varphi)$ to find the states $q \in Q$ in which $tr(\varphi)$ holds. Then we update the model, adding a new proposition p that holds exactly in these states, and we model-check $(p \wedge \langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle\bigcirc\langle\langle A \cup \mathbb{A}\text{gt}^e\rangle\rangle\Box(\mathsf{act} \wedge p))$ as the translation of $\langle\langle A\rangle\rangle\Box\varphi$ in the new model. We tackle $tr(\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi)$ likewise.

Since the complexity of transforming $\mathcal{M}$ to $\mathcal{M}^{\text{ATL}}$ is no worse than $O(n^2)$, and the complexity of ATL model checking algorithm from (Alur et al., 2002) is $O(ml)$, the interpretation defined above can be used, for instance, for an efficient reduction of model checking of ATEL$_{CE}$ formulae to model checking in ATL.

3.4.4 Interpreting Models and Formulae of Full ATEL

Now, in order to interpret the full ATEL we modify the construction by introducing new epistemic agents (and states) indexed not only with individual agents, but with all possible non-empty coalitions:

$$\mathbb{A}\mathrm{gt}^e = \{e_A \mid A \subseteq \mathbb{A}\mathrm{gt}, A \neq \varnothing\}$$
$$Q' = Q \cup \bigcup_{A \subseteq \mathbb{A}\mathrm{gt}, A \neq \varnothing} Q^{\mathsf{e_A}},$$

where Q and all $Q^{\mathsf{e_A}}$ are pairwise disjoint. Accordingly, we extend the language with new propositions $\{\mathsf{e_A} \mid A \subseteq \mathbb{A}\mathrm{gt}\}$. The choices for complex epistemic agents refer to the (epistemic copies of) states accessible via distributed knowledge relations: $\Sigma'_{e_A} = \{pass\} \cup img(q, \sim_A^D)^{\mathsf{e_A}}$. Then we modify the transition function (putting the strategies from epistemic agents in any predefined order):

$$o'_q(\sigma_{a_1}, ..., \sigma_{a_k}, ..., \sigma_{e_A}, ...) = \begin{cases} o_q(\sigma_{a_1}, ..., \sigma_{a_k}) & \text{if all } \sigma_{e_A} = pass \\ \sigma_{e_A} & \text{if } e_A \text{ is the first active epistemic agent} \end{cases}$$

Again, the game frames for all epistemic copies of the action states are the same. The translation for all operators remain the same as well (just using $e_{\{i\}}$ instead of e_i) and the translation of D_A is:

$$tr(D_A\varphi) = \neg\langle\langle \mathbb{A}\mathrm{gt}^e \rangle\rangle \bigcirc (e_A \wedge \neg tr(\varphi)).$$

The following result can now be proved similarly to Theorem 3.6.

Theorem 3.7 *For every* ATEL *formula φ and model $\mathcal{M}$, φ is satisfiable (resp. valid) in $\mathcal{M}$ iff $tr(\varphi)$ is satisfiable (resp. valid) in $\mathcal{M}^{\mathrm{ATL}}$.*

This interpretation requires (in general) an exponential blow-up of the original ATEL model (in the number of agents k). We suspect that this may be inevitable – if so, this tells something about the inherent complexity of the epistemic operators. For a specific ATEL formula φ, however, we do not have to include all the epistemic agents e_A in the model – only those for which D_A occurs in φ. Also, we need epistemic states only for these coalitions. Note that the number of such coalitions is never greater than the length of φ. Let l be the length of formula φ, and let $\overline{m}$ be the cardinality of the "densest" modal accessibility relation – either strategic or epistemic – in $\mathcal{M}$. In other words, $\overline{m} = \max(m, m_{\sim_1}, ..., m_{\sim_k})$, where m is the number of transitions in $\mathcal{M}$, and $m_{\sim_1}, ..., m_{\sim_k}$ are cardinalities of the respective epistemic relations. Then, the "optimized" transformation gives us a model with $m' = O(l \cdot \overline{m})$ transitions, while the new formula $tr(\varphi)$ is again only linearly longer than φ (in the sense explained in Section 4.4.2). In consequence, we can still use the ATL model checking algorithm for model checking of ATEL formulae that is linear in the size of the original structure: the complexity of such process is $O(\overline{m}\, l^2)$.

3.4.5 Planning for Epistemic Goals

If an agent models reality in terms of alternating epistemic transition systems, multi-player epistemic game models or epistemic coalition effectivity models, he can use the above interpretation to reduce ATEL *planning* to ATL planning as well. Thus – via the reduction – we obtain an efficient planning algorithm for goals that may include epistemic properties: an ATEL goal φ and the model of the reality $\mathcal{M}$ are translated to

$tr(\varphi)$ and $\mathcal{M}^{\text{ATL}}$, and we can execute the procedure $plan(tr(\varphi))$ from Section 2.8 on model $\mathcal{M}^{\text{ATL}}$. In this section we show a few examples of such planning.

It should be mentioned, however, that ATEL formulae of shape $\langle\langle A\rangle\rangle\Phi$ (where $\Phi \equiv \bigcirc\varphi$, $\Box\varphi$ or $\varphi\mathcal{U}\psi$). should rather *not* be read as "team A has a plan to enforce Φ" ATEL semantics implies unexpected properties for many models, which is especially evident when we understand "having a strategy" as "being in control to execute a plan". The issue is analyzed extensively in Chapter 4, and we propose two new variants of "ATL with incomplete information" (dubbed ATOL and ATEL-R) that seem to be in a better agreement with the common-sense understanding of strategies as (feasible and executable) plans. Unfortunately, model checking for both ATOL and ATEL-R is NP-hard (cf. Proposition 4.2 for ATOL, and the discussion of ATEL-R complexity in Section 4.5.5), so it cannot be solved in polynomial time (unless P=NP, of course) –

This is where ATEL may prove useful. Let us observe that ATEL formula $\langle\langle A\rangle\rangle\Phi$ is a necessary condition for A having an executable plan to enforce Φ, in the sense defined in (Lin, 2000; Doherty et al., 2001).[3] If A have such a plan, then $\langle\langle A\rangle\rangle\Phi$ must hold; thus $\neg\langle\langle A\rangle\rangle\Phi$ implies that such a plan does not exist (this observation is given a more formal treatment in Proposition 4.6-4, and Proposition 4.10-3 for ATOL and ATEL-R, respectively). In that sense, ATEL formulae can be used to specify an upper approximation of the agents' *real* abilities.

Remark 3.8 *We pointed out in Section 2.8.4 that model checking of strategic formulae from* ATL *(and hence from* ATEL *as well) comes very close to the algorithm of minimaxing in zero-sum games. In the same sense, the approximate evaluation of* ATOL *formulae through their* ATEL *necessary condition counterparts, that we suggest above, strongly resembles the technique of* Monte Carlo Sampling*) (Corlett and Todd, 1985; Ginsberg, 1996, 1999). Monte Carlo minimaxing was successfully used as a basis for GIB, the first Bridge playing program that could seriously compete with human players (Ginsberg, 1999, 2001).*

Example 3.5 Consider the server/client system from Example 3.3. Suppose we want to check whether in any state there is common knowledge among the agents about some situation being *not* the current situation. In other words: can they rule out some state as impossible to be the case at this moment, and know that the others can rule it out, and know that they know... etc. The ATEL formula to be checked is then: $C_{\{s,c\}}\neg\mathsf{x{=}0} \vee C_{\{s,c\}}\neg\mathsf{x{=}1} \vee C_{\{s,c\}}\neg\mathsf{x{=}2}$. We use the construction from Section 3.4.3 to transform the multi-player epistemic game model $\mathcal{M}$ from Figure 3.3 to obtain the corresponding model $\mathcal{M}^{\text{ATL}}$ without epistemic relations (cf. Figure 3.5). The translation gives the following ATL formula:

$$\begin{aligned}\Phi \;\equiv\;\; & \neg\langle\langle e_s, e_c\rangle\rangle\bigcirc\langle\langle e_s, e_c\rangle\rangle(\mathsf{e_s} \vee \mathsf{e_c})\,\mathcal{U}((\mathsf{e_s} \vee \mathsf{e_c}) \wedge \neg\neg\mathsf{x{=}0})\\ & \vee \neg\langle\langle e_s, e_c\rangle\rangle\bigcirc\langle\langle e_s, e_c\rangle\rangle(\mathsf{e_s} \vee \mathsf{e_c})\,\mathcal{U}((\mathsf{e_s} \vee \mathsf{e_c}) \wedge \neg\neg\mathsf{x{=}1})\\ & \vee \neg\langle\langle e_s, e_c\rangle\rangle\bigcirc\langle\langle e_s, e_c\rangle\rangle(\mathsf{e_s} \vee \mathsf{e_c})\,\mathcal{U}((\mathsf{e_s} \vee \mathsf{e_c}) \wedge \neg\neg\mathsf{x{=}2}).\end{aligned}$$

Executing $plan(\Phi)$ in the context of model $\mathcal{M}^{\text{ATL}}$ returns the empty set. Thus, $C_{\{s,c\}}\neg\mathsf{x{=}0} \vee C_{\{s,c\}}\neg\mathsf{x{=}1} \vee C_{\{s,c\}}\neg\mathsf{x{=}2}$ is true in no state of M. □

[3]We conjecture that it might be the strongest necessary condition for the property of having a feasible plan under incomplete information.

Remark 3.9 *Note that, while looking for a plan that brings about some φ, we are usually interested in getting the result for the original states from $\mathcal{M}$ only, as the epistemic states from $\mathcal{M}^{\text{ATL}}$ play only an auxiliary role in the procedure. In such a case, the formula we must use to ask whether property φ can be maintained forever, can be simplified to $\langle\!\langle A \cup \mathbb{A}\text{gt}^e \rangle\!\rangle \Box(\mathsf{act} \wedge \varphi)$, instead of the complicated $\varphi \wedge \langle\!\langle A \cup \mathbb{A}\text{gt}^e \rangle\!\rangle \bigcirc \langle\!\langle A \cup \mathbb{A}\text{gt}^e \rangle\!\rangle \Box(\mathsf{act} \wedge \varphi)$. A similar remark applies to $\langle\!\langle A \rangle\!\rangle \varphi \mathcal{U} \psi$.*

Putting it in a more formal way, we observe that the following clauses allow some simplification of the translated ATL *formulae serving as the planning procedure input:*

1. *for every $q \in Q$:*
   $\mathcal{M}^{\text{ATL}}, q \models \varphi \wedge \langle\!\langle A \cup \mathbb{A}\text{gt}^e \rangle\!\rangle \bigcirc \langle\!\langle A \cup \mathbb{A}\text{gt}^e \rangle\!\rangle \Box(\mathsf{act} \wedge \varphi)$
   iff $\mathcal{M}^{\text{ATL}}, q \models \langle\!\langle A \cup \mathbb{A}\text{gt}^e \rangle\!\rangle \Box(\mathsf{act} \wedge \varphi)$*;*

2. *for every $q \in Q$:*
   $\mathcal{M}^{\text{ATL}}, q \models \psi \vee (\varphi \wedge \langle\!\langle A \cup \mathbb{A}\text{gt}^e \rangle\!\rangle \bigcirc \langle\!\langle A \cup \mathbb{A}\text{gt}^e \rangle\!\rangle (\mathsf{act} \wedge \varphi) \mathcal{U} (\mathsf{act} \wedge \psi))$
   iff $\mathcal{M}^{\text{ATL}}, q \models \langle\!\langle A \cup \mathbb{A}\text{gt}^e \rangle\!\rangle (\mathsf{act} \wedge \varphi) \mathcal{U} (\mathsf{act} \wedge \psi)$.

Note that the above equivalences hold only for the "action" states from the original set Q, so they cannot be used to simplify the whole translation.

Example 3.6 Let us go back to the agents from Example 3.5. Common knowledge about a situation being *not* the case now does not hold in any state, so it cannot be established in the future as well. We may ask, however, if there is a way to ensure at least that *all the agents* can rule out such a situation (although they probably will not know that they can): $\langle\!\langle s, c \rangle\!\rangle \Diamond \big(E_{\{s,c\}} \neg \mathsf{x{=}0} \vee E_{\{s,c\}} \neg \mathsf{x{=}1} \vee E_{\{s,c\}} \neg \mathsf{x{=}2}\big)$. Suppose we are particularly interested in ruling out the case where $\mathsf{x{=}2}$ to simplify the example. Now, formula $\langle\!\langle s, c \rangle\!\rangle \Diamond E_{\{s,c\}} \neg \mathsf{x{=}2}$ is in fact a shorthand for $\langle\!\langle s, c \rangle\!\rangle \top \mathcal{U} E_{\{s,c\}} \neg \mathsf{x{=}2}$, which can be translated to ATL as:

$$\Phi \equiv \langle\!\langle s, c, e_s, e_c \rangle\!\rangle \mathsf{act}\, \mathcal{U} \big(\mathsf{act} \wedge \neg \langle\!\langle e_s, e_c \rangle\!\rangle \bigcirc ((\mathsf{e_s} \vee \mathsf{e_c}) \wedge \mathsf{x{=}2})\big).$$

Procedure $plan(\Phi)$ in the context of model $\mathcal{M}^{\text{ATL}}$ returns:

$$\{\langle q_0, \langle accept, add1, pass, pass \rangle \rangle, \langle q_1, - \rangle, \langle q_2, \langle accept, add1, pass, pass \rangle \rangle\}.$$

Thus, agents s and c can execute collective strategy

$$\{\langle q_0, \langle accept, add1 \rangle \rangle, \langle q_1, - \rangle, \langle q_2, \langle accept, add1 \rangle \rangle\}$$

to eventually achieve $E_{\{s,c\}} \neg \mathsf{x{=}2}$. □

3.4.6 Interpretation of ATEL* into ATL*

Although the computational complexity makes ATEL* model checking rather uninteresting from the practical standpoint, we find it worth pointing out that the translation from Section 3.4.4 is in fact a special case of a more general interpretation that enables translating ATEL* formulae into ATL*. We briefly sketch the latter interpretation here.

First, models are translated in exactly the same way as in Section 3.4.4. Second, the translation of formulae is given below:

$$
\begin{aligned}
tr(p) &= p, \qquad \text{for } p \in \Pi \\
tr(\neg\varphi) &= \neg tr(\varphi) \\
tr(\varphi \vee \psi) &= tr(\varphi) \vee tr(\psi) \\
tr(\langle\langle A\rangle\rangle\varphi) &= \langle\langle A \cup \mathbb{A}\mathrm{gt}^e\rangle\rangle tr(\varphi) \\
tr(\bigcirc\varphi) &= \bigcirc(\mathsf{act} \wedge tr(\varphi)) \\
tr(\varphi\mathcal{U}\psi) &= tr(\psi) \vee (tr(\varphi) \wedge \bigcirc(\mathsf{act} \wedge tr(\varphi))\mathcal{U}(\mathsf{act} \wedge tr(\psi))) \\
tr(K_{a_i}\varphi) &= \neg\langle\langle\{e_1, ..., e_i\}\rangle\rangle\bigcirc(\mathsf{e_i} \wedge \neg tr(\varphi)) \\
tr(E_A\varphi) &= \neg\langle\langle\mathbb{A}\mathrm{gt}^e\rangle\rangle\bigcirc(\bigvee_{a_i \in A} \mathsf{e_i} \wedge \neg tr(\varphi)) \\
tr(C_A\varphi) &= \neg\langle\langle\mathbb{A}\mathrm{gt}^e\rangle\rangle\bigcirc\big((\bigvee_{a_i \in A} \mathsf{e_i})\mathcal{U}(\bigvee_{a_i \in A} \mathsf{e_i} \wedge \neg tr(\varphi))\big) \\
tr(D_A\varphi) &= \neg\langle\langle\mathbb{A}\mathrm{gt}^e\rangle\rangle\bigcirc(e_A \wedge \neg tr(\varphi)).
\end{aligned}
$$

3.5 BDI and Its Interpretation in ATL

One of the best known frameworks for reasoning about rational agents, inspired by the philosophical debate on the nature of agency and building upon the repository of various modal logics, is the BDI logic proposed in (Rao and Georgeff, 1991, 1995) and later investigated in (Wooldridge, 2000). BDI enables expressing claims about an agent's *beliefs*, *desires* and *intentions*, wrapped around the language of computation tree logic. The original language of BDI is very ornate, including first-order version of full CTL*, plus elements of dynamic logic, quantification over agents and actions etc. The main "specialty" of the logic, however, lies in its models: the possible worlds are *not* instantaneous states of the system, but rather *computational trees* themselves (emphasizing the fact that in every situation we may see different possible lines of future), and the accessibility relations are *ternary* rather than binary (showing which possible worlds are indistinguishable *at a particular time point*).[4] In this section we will discuss BDI_{CTL}, a propositional variant of BDI, the way it was defined in (Schild, 2000). We will also follow that paper in our presentation of the logic's semantics, defining the models as more conventional structures, in which the successor relation and the accessibility relations must satisfy certain conditions.

Formulae of BDI_{CTL} are:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \mathsf{E}\bigcirc\varphi \mid \mathsf{E}\Box\varphi \mid \mathsf{E}\varphi_1\mathcal{U}\varphi_2 \mid \mathsf{Bel}_a\varphi \mid \mathsf{Des}_a\varphi \mid \mathsf{Int}_a\varphi$$

The semantics of BDI_{CTL} can be based on **situation structures**:

$$\mathcal{S} = \langle \Omega, \mathbb{A}\mathrm{gt}, \mathcal{R}, \mathcal{B}_{a_1}, ..., \mathcal{B}_{a_k}, \mathcal{D}_{a_1}, ..., \mathcal{D}_{a_k}, \mathcal{I}_{a_1}, ..., \mathcal{I}_{a_k}, \pi \rangle,$$

[4] It is worth noting that this sort of structures resembles to some extent the representation proposed independently in (Frank et al., 1998) to investigate search algorithms for games with incomplete information.

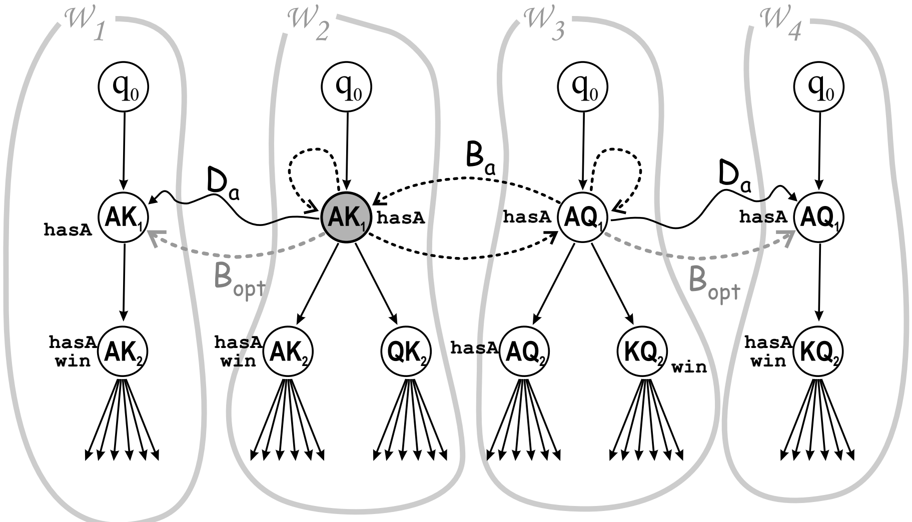


Figure 3.6: A fragment of a BDI model for the card game: a is the actual player in our game; *opt* is the "optimistic player", or at least this agent represents player a's idea about a "real optimist".

where:

- $\Omega = W \times T$ is a set of **situations**. A situation $\langle w, t \rangle$ is a pair of a *possible world* $w \in W$ and a *time point* $t \in T$ in w;
- $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$ is a set of agents;
- relation $\mathcal{R}$ defines transitions between situations. It is required that the situations belong to the same possible world: if $\langle w, t \rangle \mathcal{R} \langle w', t' \rangle$ then $w = w'$;
- relations $\mathcal{B}_a$ show which situations are considered possible by an agent a from the current situation. The situations must share the time point: if $\langle w, t \rangle \mathcal{B}_a \langle w', t' \rangle$ then $t = t'$. $\mathcal{B}_a$ is assumed serial, transitive and Euclidean;
- relations $\mathcal{D}_a$ ($\mathcal{I}_a$) show which situations are considered desirable (intended) by a. Again, if $\langle w, t \rangle \mathcal{D}_a \langle w', t' \rangle$ ($\langle w, t \rangle \mathcal{I}_a \langle w', t' \rangle$, respectively) then $t = t'$. $\mathcal{D}_a$ and $\mathcal{I}_a$ are only assumed to be serial;
- $\pi : \Omega \rightarrow \mathcal{P}(\Pi)$ is a valuation of propositions from Π.

Example 3.7 Consider a very simple card game, that will serve us as a working example in Chapter 4 (cf. Example 4.2). Agent a plays against the environment env, and the deck consists of Ace, King and Queen (A, K, Q). We assume that A beats K, K beats Q, but Q beats A.[5] First env gives a card to a, and assigns one card to itself. Then

[5] Note that this scheme closely resembles the game of *RoShamBo* or "Rock-Paper-Scissors": paper covers rock, scissors cut paper, but rock crushes scissors.

a can exchange his card for the one remaining in the deck, or he can keep the current one. The player with the better card wins the game. The game is played repeatedly *ad infinitum*. Figure 3.6 presents a fragment of a (possible) BDI structure for player a; we assume that a has just been given A, env holds K, and a can keep the Ace or exchange it for the remaining card in the next move. Thus, the situation is $\langle w_2, AK_1\rangle$ now. Proposition hasA can be used to identify the situations in which player a has the Ace, and propositions win labels the winning positions for a in the game. Here are some things that can be said about the game and the player in situation $\langle w_2, AK_1\rangle$:

1. $\mathsf{hasA} \wedge \mathsf{Bel}_a\mathsf{hasA}$: agent a has the Ace, and he is aware of it;

2. $\mathsf{Bel}_a\mathsf{E}\bigcirc\mathsf{win}$: he believes there is a way to win in the next step;

3. $\mathsf{Des}_a\mathsf{A}\bigcirc\mathsf{win}$: he desires that every path leads to a victory, so he does not have to worry about his decisions. However, he does not believe it is possible, so his desires are rather unrealistic: $\neg\mathsf{Bel}_a\mathsf{A}\bigcirc\mathsf{win}$;

4. moreover, he believes that a real optimist would believe that the victory is inevitable: $\mathsf{Bel}_a\mathsf{Bel}_{opt}\mathsf{A}\bigcirc\mathsf{win}$.

□

BDI$_{CTL}$ was shown to be subsumed by the propositional *μ-calculus*, one of the standard logics of concurrency (Schild, 2000). We will use a similar technique to reduce the model checking problem BDI$_{CTL}$ to ATL and CTL. We conjecture that the interpretation might yield analogous subsumption result, i.e. that BDI$_{CTL}$ can be reduced to ATL or even CTL. At this moment, however, no definitive results have been obtained in this respect.

3.5.1 An Interpretation of BDI$_{CTL}$ into ATL and CTL

Using a construction similar to the one in Section 3.4.3, we will interpret BDI$_{CTL}$ into ATL.

Given a situation structure $\mathcal{S} = \langle\Omega, \mathcal{R}, \mathcal{B}_{a_1}, ..., \mathcal{B}_{a_k}, \mathcal{D}_{a_1}, ..., \mathcal{D}_{a_k}, \mathcal{I}_{a_1}, ..., \mathcal{I}_{a_k}, \pi\rangle$ for a set of agents $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$ over a set of propositions Π, we construct a game model $\mathcal{M}^{\mathrm{ATL}} = \langle Q', \gamma', \pi'\rangle$ as follows:

- $\mathbb{A}\mathrm{gt}' = \{env\} \cup \mathbb{A}\mathrm{gt}^{bel} \cup \mathbb{A}\mathrm{gt}^{des} \cup \mathbb{A}\mathrm{gt}^{int}$. As BDI is not really about what outcomes can be effected by which agents, the dynamic structure of the system can be attributed to one agent (the "environment") without any loss of generality. However, we need additional agents $\mathbb{A}\mathrm{gt}^{bel} = \{bel_1, ..., bel_k\}$, $\mathbb{A}\mathrm{gt}^{des} = \{des_1, ..., des_k\}$ and $\mathbb{A}\mathrm{gt}^{int} = \{int_1, ..., int_k\}$ to translate BDI formulae referring to agents' beliefs, desires and intentions;

- $Q' = \Omega \cup \bigcup(\Omega^{\mathsf{bel_i}} \cup \Omega^{\mathsf{des_i}} \cup \Omega^{\mathsf{int_i}})$, where $S^{\mathsf{bel_i}} = \{q^{\mathsf{bel_i}} \mid q \in S\}, S \subseteq \Omega$ etc. Additional states $\Omega^{\mathsf{bel_i}}$, $\Omega^{\mathsf{des_i}}$ and $\Omega^{\mathsf{int_i}}$ are copies of the original ones, and will be used to simulate the $\mathcal{B}_{a_i}$, $\mathcal{D}_{a_i}$ and $\mathcal{I}_{a_i}$ accessibility relations;

- $\Pi' = \Pi \cup \{\mathsf{act}, \mathsf{bel_1}, ..., \mathsf{bel_k}, \mathsf{des_1}, ..., \mathsf{des_k}, \mathsf{int_1}, ..., \mathsf{int_k}\}$. $\pi'(p) = \pi(p) \cup \bigcup_{i=1,...,k} (\pi(p)^{\mathsf{bel_i}} \cup \pi(p)^{\mathsf{des_i}} \cup \pi(p)^{\mathsf{int_i}})$ for every $p \in \Pi$; $\pi'(\mathsf{act}) = \Omega$, $\pi'(\mathsf{bel_i}) = \Omega^{\mathsf{bel_i}}$, $\pi'(\mathsf{des_i}) = \Omega^{\mathsf{des_i}}$ and $\pi'(\mathsf{int_i}) = \Omega^{\mathsf{int_i}}$.

For every situation q in $\mathcal{S}$, we translate the relations $\mathcal{R}$, $\mathcal{B}$, $\mathcal{D}$ and $\mathcal{I}$ into a game frame $\gamma'(q) = \langle \mathbb{A}\mathrm{gt}', \{\Sigma'_a \mid a \in \mathbb{A}\mathrm{gt}'\}, o', Q' \rangle$:

- $\Sigma'_{env} = img(q, \mathcal{R})$;
- $\Sigma'_{bel_i} = \{pass\} \cup img(q, \mathcal{B}_{a_i})^{\mathsf{bel_i}}$, $\Sigma'_{des_i} = \{pass\} \cup img(q, \mathcal{D}_{a_i})^{\mathsf{des_i}}$, and $\Sigma'_{int_i} = \{pass\} \cup img(q, \mathcal{I}_{a_i})^{\mathsf{int_i}}$ for $i = 1, ..., k$;
- again, the new transition function is defined as follows: if all the "dummy" agents decide to be passive, then a "real" transition is executed; otherwise, the choice of the first non-passive "dummy" agent is accepted:

$$o'_q(\sigma_{env}, \sigma_{bel_1}, ..., \sigma_{int_k}) = \begin{cases} \sigma_{env} & \text{if } \sigma_{bel_1} = ... = \sigma_{int_k} = pass \\ s^{\mathsf{e_i}} & \text{if } e_i \text{ is the first active agent} \\ & \text{out of } bel_1, ..., int_k, \text{ and } s = \sigma_{e_i} \end{cases}$$

The game frames for the new states are exactly the same: $\gamma'(q^{\mathsf{bel_i}}) = \gamma'(q^{\mathsf{des_i}}) = \gamma'(q^{\mathsf{int_i}}) = \gamma'(q), fori = 1, ..., k, q \in \Omega$.

Now, we define a translation of the BDI$_{CTL}$ formulae into ATL. The translation is very similar to the one for ATEL formulae, once we recall that A can be expressed with $\langle\!\langle \varnothing \rangle\!\rangle$ in ATL:

$$\begin{aligned}
tr(p) &= p \\
tr(\neg\varphi) &= \neg tr(\varphi) \\
tr(\varphi \vee \psi) &= tr(\varphi) \vee tr(\psi) \\
tr(\mathsf{E}\bigcirc\varphi) &= \langle\!\langle \{env\} \cup \mathbb{A}\mathrm{gt}^{bel} \cup \mathbb{A}\mathrm{gt}^{des} \cup \mathbb{A}\mathrm{gt}^{int} \rangle\!\rangle \bigcirc (\mathsf{act} \wedge tr(\varphi)) \\
tr(\mathsf{E}\Box\varphi) &= tr(\varphi) \wedge \langle\!\langle \{env\} \cup \mathbb{A}\mathrm{gt}^{bel} \cup \mathbb{A}\mathrm{gt}^{des} \cup \mathbb{A}\mathrm{gt}^{int} \rangle\!\rangle \bigcirc \\
&\qquad \langle\!\langle \{env\} \cup \mathbb{A}\mathrm{gt}^{bel} \cup \mathbb{A}\mathrm{gt}^{des} \cup \mathbb{A}\mathrm{gt}^{int} \rangle\!\rangle \Box (\mathsf{act} \wedge tr(\varphi)) \\
tr(\mathsf{E}\varphi\,\mathcal{U}\psi) &= tr(\psi) \vee tr(\varphi) \wedge \langle\!\langle \{env\} \cup \mathbb{A}\mathrm{gt}^{bel} \cup \mathbb{A}\mathrm{gt}^{des} \cup \mathbb{A}\mathrm{gt}^{int} \rangle\!\rangle \bigcirc \\
&\qquad \langle\!\langle \{env\} \cup \mathbb{A}\mathrm{gt}^{bel} \cup \mathbb{A}\mathrm{gt}^{des} \cup \mathbb{A}\mathrm{gt}^{int} \rangle\!\rangle (\mathsf{act} \wedge tr(\varphi))\,\mathcal{U}(\mathsf{act} \wedge tr(\psi)) \\
tr(\mathsf{Bel}_{a_i}\varphi) &= \neg\langle\!\langle \{bel_1, ..., bel_i\} \rangle\!\rangle \bigcirc (\mathsf{bel_i} \wedge \neg tr(\varphi)) \\
tr(\mathsf{Des}_{a_i}\varphi) &= \neg\langle\!\langle \mathbb{A}\mathrm{gt}^{bel} \cup \{des_1, ..., des_i\} \rangle\!\rangle \bigcirc (\mathsf{des_i} \wedge \neg tr(\varphi)) \\
tr(\mathsf{Int}_{a_i}\varphi) &= \neg\langle\!\langle \mathbb{A}\mathrm{gt}^{bel} \cup \mathbb{A}\mathrm{gt}^{des} \cup \{int_1, ..., int_i\} \rangle\!\rangle \bigcirc (\mathsf{int_i} \wedge \neg tr(\varphi))
\end{aligned}$$

Theorem 3.10 *For every* BDI$_{CTL}$ *formula* ϕ *and model* $\mathcal{M}$*,* ϕ *is satisfiable (resp. valid) in* $\mathcal{M}$ *iff* $tr(\phi)$ *is satisfiable (resp. valid) in* $\mathcal{M}^{\mathrm{ATL}}$.

The proof is analogous to Theorem 3.6.

Note that $tr(\mathsf{Bel}_{a_i}\varphi)$ may be as well rephrased as

$$\neg\langle\!\langle \mathbb{A}\mathrm{gt} \cup \mathbb{A}\mathrm{gt}^{bel} \cup \mathbb{A}\mathrm{gt}^{des} \cup \mathbb{A}\mathrm{gt}^{int} \rangle\!\rangle \bigcirc (\mathsf{bel_i} \wedge \neg tr(\varphi))$$

since it does not matter what the agents from $\mathbb{A}\mathrm{gt} \cup \mathbb{A}\mathrm{gt}^{des} \cup \mathbb{A}\mathrm{gt}^{int} \cup \{bel_{i+1}, ..., bel_k\}$ decide to do at all. Note also that $\langle\langle \mathbb{A}\mathrm{gt} \cup \mathbb{A}\mathrm{gt}^{bel} \cup \mathbb{A}\mathrm{gt}^{des} \cup \mathbb{A}\mathrm{gt}^{int} \rangle\rangle$ is equivalent to the existential path quantifier E from CTL. Similar remarks apply to the rest of above clauses. After re-writing the clauses, we obtain an equivalent translation:

$$
\begin{aligned}
tr(p) &= p \\
tr(\neg\varphi) &= \neg tr(\varphi) \\
tr(\varphi \vee \psi) &= tr(\varphi) \vee tr(\psi) \\
tr(\mathsf{A}\bigcirc\varphi) &= \mathsf{E}\bigcirc(\mathsf{act} \wedge tr(\varphi)) \\
tr(\mathsf{A}\Box\varphi) &= tr(\varphi) \wedge \mathsf{E}\bigcirc\mathsf{E}\Box(\mathsf{act} \wedge tr(\varphi)) \\
tr(\mathsf{A}\varphi\,\mathcal{U}\psi) &= tr(\psi) \vee tr(\varphi) \wedge \mathsf{E}\bigcirc\mathsf{E}(\mathsf{act} \wedge tr(\varphi))\,\mathcal{U}(\mathsf{act} \wedge tr(\psi)) \\
tr(\mathsf{Bel}_{a_i}\varphi) &= \neg\mathsf{E}\bigcirc(\mathsf{bel_i} \wedge \neg tr(\varphi)) \\
tr(\mathsf{Des}_{a_i}\varphi) &= \neg\mathsf{E}\bigcirc(\mathsf{des_i} \wedge \neg tr(\varphi)) \\
tr(\mathsf{Int}_{a_i}\varphi) &= \neg\mathsf{E}\bigcirc(\mathsf{int_i} \wedge \neg tr(\varphi))
\end{aligned}
$$

Now, we can "flatten" the model $\mathcal{M}^{\mathrm{ATL}}$, leaving only the bare temporal structure (i.e. states and unlabeled transitions only) – and finally we end up with an interpretation of BDI_{CTL} into CTL itself.

3.6 Final Remarks

The satisfiability preserving interpretations of ATEL and BDI_{CTL} models and formulae into ATL models and formulae constitute the main results of this chapter in the technical sense. It was already pointed out in (van Otterloo et al., 2003) that such an interpretation portrays knowledge as a strategic ability of a special kind – which seems a nice and potent metaphor. Moreover, it allows to use existing model checking tools like MOCHA (Alur et al., 2000) for an efficient model checking of ATEL and BDI_{CTL}. The aim of this study goes beyond the formal claims being presented, though. We wanted to show the logics of ATEL and BDI as parts of a bigger picture, so that one can compare them, appreciate their similarities and differences, and choose the system most suitable for the intended application.

The picture suggests that BDI and ATL/ATEL can contribute to each other, too:

- The BDI notions of *desire* and *intention* can enrich ATEL directly, both on the syntactical an semantic level.
- ATL and coalition games can provide BDI models with a finer-grained structure of action (simultaneous choices). Furthermore, the cooperation modalities can be "imported" into the BDI framework to enable modeling, specifying and verifying agents' strategic abilities in the context of their beliefs, desires and intentions.
- The treatment of group epistemics from ATEL can be used in the BDI logics too.

Chapter 4

Agents that Know how to Play

SYNOPSIS. *We have been looking for a good formal language to model and reason about agents for the last two chapters. We settled for the logics of "strategic ability": most notably, Alternating-time Temporal Logic. Then came Alternating-time Temporal Epistemic Logic (*ATEL*) that endowed our agents with the ability to be uncertain. And to have some knowledge. Plus, endowed us with the ability to model and reason about agents' knowledge and uncertainty. A seemingly perfect framework for talking about what agents can and cannot achieve under incomplete information.*

We already hinted that interpreting strategic ability of ATEL *agents as "being in control to execute a plan" can be misleading. Now the time has come to look at this issue more carefully. The tension grows. Are* ATEL *models and formulae what they seem?*

4.1 Introduction

The logics of ATL and CL, investigated in Chapter 2, offer an intuitively appealing perspective to systems inhabited by autonomous entities. However, they refer only to agents who have perfect and complete information about the world they live in, which is somewhat unrealistic. We strongly believe that Alternating-time Epistemic Logic, introduced in (van der Hoek and Wooldridge, 2002) and discussed at length in Chapter 3, is a move in the right direction. ATEL adds to ATL the vocabulary of epistemic logic, with its long tradition of talking about agents' knowledge under incomplete information. Still, in ATEL the strategic and epistemic layers are combined as if they were independent. They are – if we do not ask whether the agents in question are able to identify and execute their strategies. They are not if we want to interpret strategies as *feasible plans* that guarantee achieving the goal. This issue is going to be discussed in this chapter.

One of the main challenges in ATEL, not really addressed in (van der Hoek and Wooldridge, 2002) but already hinted upon in (Jamroga, 2003d), is the question how, given an explicit way to represent the agent's knowledge, this should interfere with the agents' available strategies. What does it mean that an agent has a way to enforce φ, if he should therefore make different choices in epistemically indistinguishable states, for instance? In Section 4.3, we argue that in order to add an epistemic component to ATL, one should give an account of the tension between *incomplete information* that is imposed on the agents on one hand, and *perfect recall* that is assumed about them when they are to make their decisions, on the other hand. We also argue that, when reasoning about what an agent can *enforce*, it seems more appropriate to require the agent knows his winning strategy rather than he knows only that such a strategy exists.

Then, in Section 4.4 we will loosen the assumption of perfect recall to agents having no, or only limited memory. The epistemic component in Alternating-time Temporal Observational Logic (ATOL) is entirely based on the notion of observation: the agents can recall no history of the game except of the information "stored" in their local states. We give several examples of what agents can achieve if they are allowed to make specific observations. Then, in Section 4.5, full Alternating-time Temporal Epistemic Logic with Recall (ATEL-R*) is considered; here, agents are again allowed to memorize the whole game. We propose a semantics for ATEL-R*, and we use past-time operators to relate the several epistemic modalities; finally, expressivity and complexity of ATEL-R* is briefly investigated.

This chapter presents research from (Jamroga and van der Hoek, 2004), a paper co-written with Wiebe van der Hoek from the University of Liverpool.

4.2 Prelude: Unraveling AETS and a Look at Strategies

Going from the model to the behavioral structure behind it, there are at least two ways of unraveling the alternating epistemic transition system into a computation tree with epistemic relations. If agents have no recall of the past, except for the information encapsulated in the current state (modulo relation $\sim$), then only the last state in a sequence matters for the epistemic accessibility links; if the agents can remember the history of previous states, then the whole sequence matters: the agents cannot discriminate two situations if they cannot distinguish any corresponding parts from the alternative histories.

Example 4.1 *Let us consider the variable client/server alternating epistemic transition system from Example 3.2 (cf. Figure 3.2). Both ways of unraveling the AETS are presented in Figure 4.1 (A and B).*

These two approaches reflect in fact two different "common-sense" interpretations of the computational structure with an epistemic component. In (A), a state (together with relation $\sim_a$) is meant to constitute the whole description of an agents' position, while in (B) states (and $\sim_a$) are more about what agents can perceive or observe at that point. More precisely, since agent c cannot distinguish q_0 from q_0q_0 in Figure 4.1A, he is not aware of any transition being happened that stays in q_0. In Figure 4.1B however, indistinguishable situations occur always on the same level of the tree, denoting that here the agents at least know how many transitions have been made.

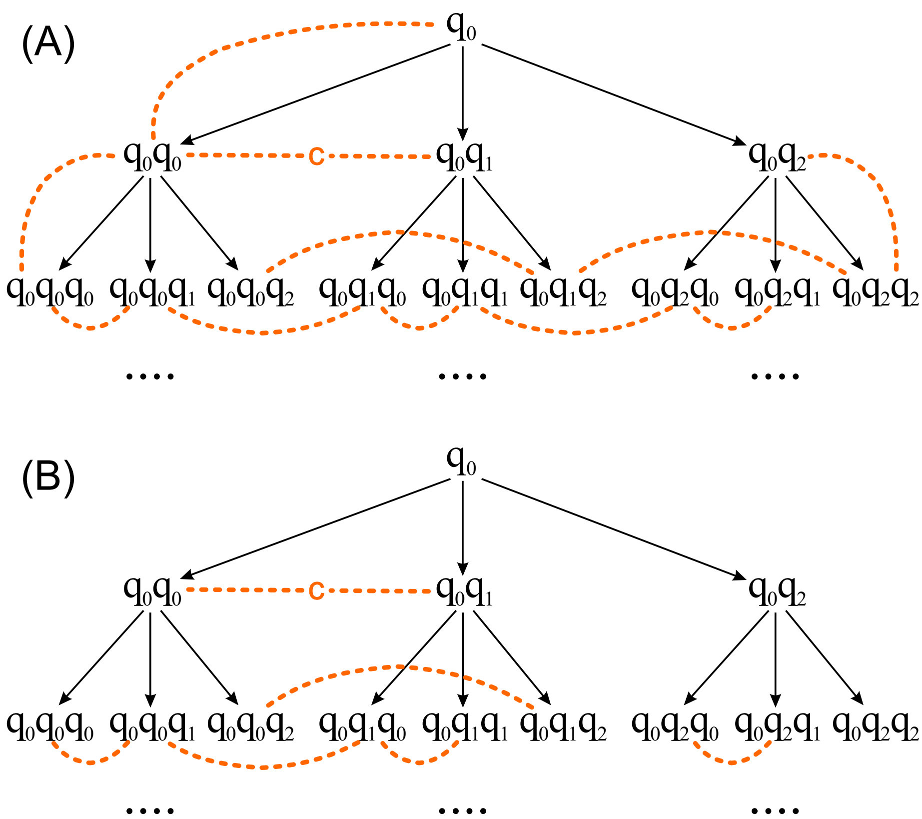


Figure 4.1: Unraveling: the computation trees with an epistemic relation for the client process. (A) indistinguishability relation based completely on $\sim_c$ — the agent does not remember the history of the game; (B) the client has perfect recall. The resulting indistinguishability relation should be read as the reflexive and transitive closure of the dotted arcs.

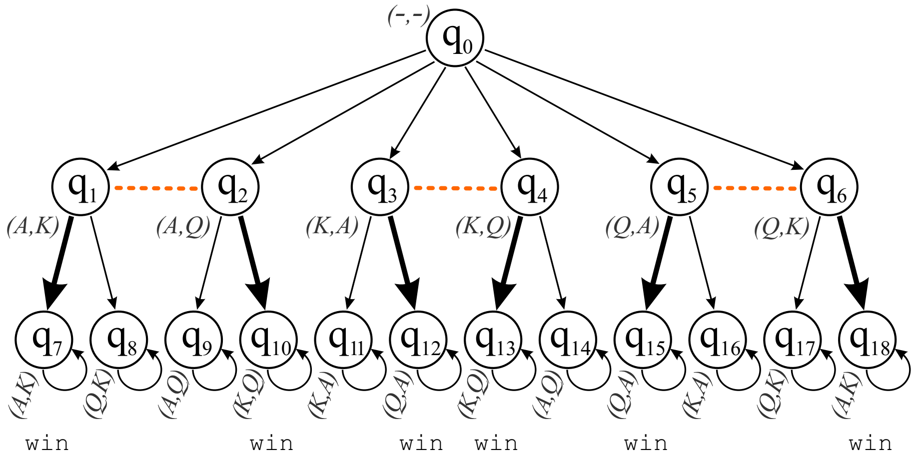
Note that ATEL agents are assumed to have perfect recall within the semantics of cooperation modalities: the knowledge available to agent a when he is choosing his action is determined by the type of strategy function f_a (which allows a to remember the whole history of previous states). Thus the epistemic abilities of agents with respect to their decision making should be the ones shown in Figure 4.1B. On the other hand, the knowledge modality K_a refers to indistinguishability of *states* — therefore its characteristics is rather displayed in Figure 4.1A.

Let us go back to the AETS from Figure 3.2. It is easy to observe that $\neg x = 2 \rightarrow \langle\langle c \rangle\rangle \bigcirc x = 2$ is valid in the system (because $q_0 \models \langle\langle c \rangle\rangle \bigcirc x = 2$ and $q_1 \models \langle\langle c \rangle\rangle \bigcirc x = 2$), which is counterintuitive: c cannot really choose a good strategy to enforce $\bigcirc x = 2$ since he can never be sure whether the system is in q_0 or q_1. Asking about c's knowledge does not make things better: it can be proved that $K_c(\neg x = 2 \rightarrow \langle\langle c \rangle\rangle \bigcirc x = 2)$, too. As it turns out, not every function of type $f_a : Q^+ \rightarrow 2^Q$ represents a feasible strategy under incomplete information. We will study the problem in more detail throughout the next section.

4.3 Knowledge and Action under Uncertainty

ATEL and ATEL* are interesting languages to describe and verify properties of autonomous processes in situations of incomplete information. However, their semantics — the way it is defined in (van der Hoek and Wooldridge, 2002) — is not entirely consistent with the assumption that agents have incomplete information about the current state. Something seems to be lacking in the definition of a valid strategy for an agent in AETS. When defining a strategy, the agent can make his choices for every state independently. This is not feasible in a situation of incomplete information if the strategy is supposed to be deterministic: if a cannot recognize whether he is in situation s_1 or s_2, he cannot plan to proceed with one action in s_1, and another in s_2. Going back to Example 3.2, since the client cannot epistemically distinguish q_0 from q_1, and in both he should apply a different strategy to ensure that x will have the value of 2 in the next state, it is not realistic to say that the client has a strategy to enforce $\bigcirc(x = 2)$ in q_0. It is very much like with the information sets from von Neumann and Morgenstern (von Neumann and Morgenstern, 1944): for every state in an information set the same action must be chosen within a strategy. Such strategies are sometimes called *uniform* in the field of logic and games (van Benthem, 2001, 2002).

Example 4.2 Consider the following example: agent a plays a very simple card game against the environment env. The deck consisting of Ace, King and Queen (A, K, Q); it is assumed that A beats K, K beats Q, but Q beats A. First env gives a card to a, and assigns one card to itself. Then a can trade his card for the one remaining in the deck, or he can keep the current one. The player with the better card wins the game. A turn-based synchronous AETS for the game is shown in Figure 4.2. Right after the cards are given, a does not know what is the hand of the other player; for the rest of the game he has complete information about the state. Atomic proposition `win` enables to recognize the states in which a is the winner. States $q_7, ..., q_{18}$ are the final states for the this game; however, the transition function *must* specify at least one outgoing



$\mathbb{A}\mathrm{gt} = \{a, env\}$
$\delta(q_0, a) = \{\{q_1, ..., q_6\}\}$
$\delta(q_1, a) = \{\{q_7\}, \{q_8\}\}$ etc.
$\delta(q_0, env) = \{\{q_1\}, ..., \{q_6\}\}$
$\delta(q_1, env) = \{\{q_7, q_8\}\}$ etc.
$\delta(q_7, a) = \delta(q_7, env) = \{\{q_7\}\}$ etc.

$q_0 \sim_a q_0$
$q_1 \sim_a q_1,\ q_1 \sim_a q_2,\ q_2 \sim_a q_2$
$q_3 \sim_a q_3,\ q_3 \sim_a q_4,\ q_4 \sim_a q_4$ etc.
$q_7 \sim_a q_7,\ q_8 \sim_a q_8,\ q_9 \sim_a q_9$ etc.
$q_0 \sim_{env} q_0,\ q_1 \sim_{env} q_1,\ q_2 \sim_{env} q_2$ etc.

Figure 4.2: Epistemic transition system for the card game. For every state, the players' hands are described. The dotted lines show a's epistemic accessibility relation $\sim_a$. The thick arrows indicate a's winning strategy.

transition for each state. A reflexive arrow at every final state shows that – once the game is over – the system remains in that state forever.

Note that $q_0 \models \langle\langle a \rangle\rangle \Diamond \mathtt{win}$, although it should definitely be false for this game. Of course, a may *happen* to win, but he does not have the *power* to bring about winning because he has no way of recognizing the right decision until it is too late. Even if we ask about whether the player can *know* that he has a winning strategy, it does not help: $K_a \langle\langle a \rangle\rangle \Diamond \mathtt{win}$ is satisfied in q_0, too, because for all $q \in Q$ such that $q_0 \sim_a q$ we have $q \models \langle\langle a \rangle\rangle \Diamond \mathtt{win}$. □

This calls for a constraint like the one from (von Neumann and Morgenstern, 1944): if two situations s_1 and s_2 are indistinguishable, then a strategy f_a must specify the same action for both s_1 and s_2. In order to accomplish this, some relation of "subjective unrecognizability" over the agents' choices can be useful — to tell which decisions will be considered the same in which states. Probably the easiest way to accomplish this is to provide the decisions with explicit labels — the way it has been done in concurrent game structures — and assume that the choices with the same label represent the same action from the agent's subjective point of view. This kind of solution fits also well in the tradition of game theory. Note that it is harder to specify this requirement if we identify agents' actions with their outcomes completely, because the same action started in two different states seldom generates the same result. If a trades his Ace in q_1, the system moves to q_8 and a loses the game; if he trades the card in q_2, the system moves to q_{10} and he wins. Still a cannot discriminate trading the Ace in both situations.

4.3.1 Towards a Solution

The first attempt to solve the problem sketched above has been presented in (Jamroga, 2003d). The idea was to define ATEL models as concurrent game structures extended with epistemic accessibility relations:

$$S = \langle k, Q, \Pi, \pi, \sim_1, ..., \sim_k, d, o\rangle$$

where agents had the same choices available in indistinguishable states, i.e. for every q, q' such that $q \sim_a q'$ it was required that $d_a(q) = d_a(q')$ (otherwise a could distinguish q from q' by the decisions he could make).[1] An *incomplete information strategy* – we will follow (van Benthem, 2001, 2002) and call it a *uniform* strategy in this thesis – was a function $f_a : Q^+ \to \mathbb{N}$ for which the following constraints held:

- $f_a(\lambda) \leq d_a(q)$, where q was the last state in sequence λ;
- if two histories are indistinguishable $\lambda \approx_a \lambda'$ then a could not specify different choices for λ and λ' within one strategy f, i.e. $f_a(\lambda) = f_a(\lambda')$.

Two histories are indistinguishable for a if he cannot distinguish their corresponding states. Recall that the ith position of λ is denoted by $\lambda[i]$. Then $\lambda \approx_a \lambda'$ iff $\lambda[i] \sim_a \lambda'[i]$ for every i. Alternatively, decisions can be specified for sequences of *local* states instead of global ones – $f_a : Q_a^+ \to \mathbb{N}$, where local states are defined as the equivalence classes of relation $\sim_a$, i.e. $Q_a = \{[q]_{\sim_a} \mid q \in Q\}$. This kind of presentation has been employed in (Schobbens, 2003), for example.

Example 4.3 A new model for the card game is shown in Figure 4.3. Now, using only uniform strategies, a is unable to bring about winning on his own: $q_0 \models \neg\langle\langle a\rangle\rangle\Diamond win$. Like in the real game, he can win only with some "help" from the environment: $q_0 \models \langle\langle a, env\rangle\rangle\Diamond win$. □

Unfortunately, the new constraint proves insufficient for ruling out strategies that are not feasible under incomplete information. Consider the last game structure and state q_1. It is easy to show that $q_1 \models \langle\langle a\rangle\rangle\Diamond win$. Moreover, $q_0 \models \langle\langle\rangle\rangle\bigcirc\langle\langle a\rangle\rangle\Diamond win$, although still $q_0 \not\models \langle\langle a\rangle\rangle\Diamond win$. In other words, no conditional plan is possible for a at q_0, and at the same time he is bound to have one in the next step! The paradoxical results lead in fact to one fundamental question: *what does it mean for an agent to have a plan*?

[1] The authors of ATEL suggested a similar requirement in (van der Hoek and Wooldridge, 2003b). They also considered whether some further constraint on the possible runs of the system should be added, but they dismissed the idea.

$\mathbb{A}\text{gt} = \{a, env\}$
$Act = \{deal_{AK}, deal_{AQ}, deal_{KA}, deal_{KQ}, deal_{QA}, deal_{QK}, trade, keep, wait\}$

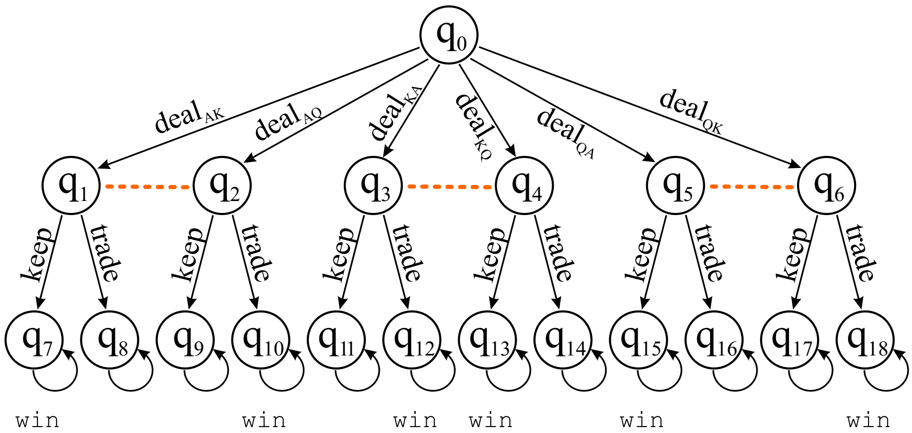


Figure 4.3: New model for the game. The transitions are labeled with decisions from the player who takes turn.

4.3.2 Having a Strategy: *de re* vs. *de dicto*

The consecutive attempts to ATEL semantics seem to refer to various levels of "strategic" nondeterminism:

1. the first semantics proposed in (van der Hoek and Wooldridge, 2002) allows for *subjectively non-deterministic strategies* in a sense: the agent is allowed to guess which choice is the right one, and if there is any way for him to guess correctly, we are satisfied with this. Therefore the notion of a strategy from (van der Hoek and Wooldridge, 2002) makes formula $\langle\langle A \rangle\rangle \Phi$ describe what coalition A may *happen* to bring about against the most efficient enemies (i.e. when the enemies know the current state and even A's collective strategy beforehand), whereas the original idea from ATL was rather about A being able to *enforce* Φ;

2. in the updated semantics from (Jamroga, 2003d), presented in the previous section, every strategy is deterministic (i.e. uniform), but the agent can choose non-deterministically between them (guess which one is right). This is because $\langle\langle a \rangle\rangle \Phi$ (in the updated version) is true if there is a consistent way of enforcing Φ, but the agent may be unaware of it, and unable to obtain it in consequence;

3. we can strengthen the condition by requiring that $K_a \langle\langle a \rangle\rangle \Phi$: still, this is not enough as the examples showed. For every q indistinguishable from the current state, a must have a strategy to achieve Φ from q – but these can be different strategies for different q's. Thus, $K_a \langle\langle a \rangle\rangle \Phi$ (in the updated version) is true if a

knows there is a consistent way of enforcing Φ – unfortunately he is not required to know the way itself;

4. for planning purposes, the agent should be rather interested in having a strategy and *knowing it* (i.e. not only knowing that he has *some* strategy).

The above hierarchy reminds the distinction between beliefs *de re* and beliefs *de dicto*. The issue is well known in the philosophy of language (Quine, 1956), as well as research on the interaction between knowledge and action (Moore, 1985; Morgenstern, 1991; Wooldridge, 2000). Suppose we have dynamic logic-like modalities, parameterized with strategies: $[F_A]\Phi$ meaning "A can use strategy F_A to bring about Φ" (or: "every execution of F_A guarantees Φ"). Suppose also that strategies are required to be uniform. Cases (2), (3) and (4) above can be then described as follows:

- $\exists_{F_a}[F_a]\Phi$ is (possibly unaware) having a uniform strategy to achieve Φ (2);
- $K_a\exists_{F_a}[F_a]\Phi$ is having a strategy *de dicto* (3);
- $\exists_{F_a}K_a[F_a]\Phi$ is having a strategy *de re* (4).

This would be a flexible way to express such subtleties. However – having extended ATEL this way – we would enable explicit quantification over strategies in the object language, and the resulting logic would be propositional no more. Instead, we can change the range of computations that are taken into account by the player when analyzing a strategy — out^* must include all the (infinite) paths that are possible from the agent's subjective perspective. Since strategies in ATEL are perfect recall strategies, the player must be able to use the information from the past during his analysis of possible future courses of action. Thus, the past history is relevant for determining the set of potential outcome paths for a strategy, and it plays an important role in the definition of out^*. Section 4.3.3 offers a more detailed discussion of this issue.

We need some terminology. Let λ be a variable over finite sequences of states, and let Λ denote an infinite sequence. Moreover, for any sequence $\xi = q_0q_1\ldots$ (be it either finite or infinite):

- $\xi[i] = q_i$ is the ith position in ξ,
- $\xi_{|i} = q_0q_1\ldots q_i$ denotes the first $i+1$ positions of ξ,
- $\xi^i = q_iq_{i+1}\ldots$ is the ith suffix of ξ.

If i is greater than the length of $\xi + 1$, these notions are undefined. The length $\ell(\lambda)$ of a finite sequence λ is defined in a straightforward way.

Definition 4.1 *Let λ be a finite non-empty sequence of states, and f_a a strategy for agent a. We say that Λ is a* feasible *computation run given finite history λ and agent a's strategy f_a, if the following holds:*

- *Λ starts with a sequence indistinguishable from λ, i.e. $\Lambda_{|n} \approx_a \lambda$, where $n = \ell(\lambda) - 1$,*

- Λ *is consistent with* f_a*. In fact, only the future part of* Λ *must be consistent with* f_a *since the past-oriented part of the strategy is irrelevant: no agent can plan the past.*

Then, we define $out^*(\lambda, f_a) = \{\Lambda| \ \Lambda$ *is feasible, given* λ *and* $f_a\}$

If cooperation modalities are to reflect the property of having a strategy *de re*, then $out^*(\lambda, f_a)$ should replace the original set of objectively possible computations in the semantics of $\langle\langle a\rangle\rangle$, so that $\langle\langle a\rangle\rangle\Phi$ holds for a history λ iff there is an incomplete information strategy f_a such that Φ is true for every computation $\Lambda \in out^*(\lambda, f_a)$. Then the new semantics of the cooperation modality can be given as:

$\lambda \models \langle\langle a\rangle\rangle_{K(a)}\Phi$ iff a has a uniform strategy f_a such that for every $\Lambda \in out^*(\lambda, f_a)$ we have that Φ holds in Λ.

We use notation $\langle\langle a\rangle\rangle_{K(a)}$ to emphasize that these cooperation modalities differ from the original ones (Alur et al., 2002; van der Hoek and Wooldridge, 2002): agent a must have a uniform strategy and be able to identify it himself.

Example 4.4 Let us consider the card game example from Figure 4.3 again. Suppose q_0 has been the initial state and the system has moved to q_1 now, so the history is $\lambda = q_0q_1$. For every strategy f_a:

$$\begin{aligned} out^*(q_0q_1, f_a) &= \{\Lambda \mid \Lambda \text{ starts with } \lambda' \approx_a q_0q_1 \text{ and } \Lambda \text{ is consistent with } f_a\} \\ &= \{\Lambda \mid \Lambda \text{ starts with } q_0q, q \sim_a q_1 \text{ and } \Lambda \text{ is consistent with } f_a\} \\ &= \{\Lambda \mid \Lambda \text{ starts with } q_0q_1 \text{ or } q_0q_2 \text{ and } \Lambda \text{ is consistent with } f_a\}. \end{aligned}$$

Note that f_a must be a uniform strategy - in particular, $f_a(q_0q_1) = f_a(q_0q_2)$. There are two possible combinations of decisions for these histories:

(1) $f_1(q_0q_1) = f_1(q_0q_2) = keep$,
(2) $f_2(q_0q_1) = f_2(q_0q_2) = trade$.

Suppose there exists f such that for every $\lambda \in out^*(q_0q1, f)$ we have $\Diamond win$. We can check both cases:

case (1): $out^*(q_0q_1, f_1) = \{q_0q_1q_7q_7..., q_0q_2q_9q_9...\}$,
case (2): $out^*(q_0q_1, f_2) = \{q_0q_1q_8q_8..., q_0q_2q_1q_1...\}$.

Now, $\Diamond win$ does not hold for $q_0q_2q_9q_9...$ nor $q_0q_1q_8q_8...$, so $q_0q_1 \not\models \langle\langle a\rangle\rangle_{K(a)}\Diamond win$. □

Note also that function out^* has a different type than the old function out, and that we interpreted formula $\langle\langle a\rangle\rangle_{K(a)}\Diamond win$ over a (finite) path and not a state in the above example. This shows another very important issue: epistemic properties of alternating-time systems with perfect recall are properties of *sequences of states* rather than single states.

4.3.3 Knowledge as Past-Related Phenomenon

Throughout the preceding sections the term "situation" was used in many places instead of "state". This was deliberate. The implicit assumption that states characterize epistemic properties of agents (expressed via the semantics of knowledge operators K_a, C_A etc. in the original version of ATEL) is probably one of the confusion sources about the logic. In concurrent game structures a *state* is not a complete description of a *situation* when the agent can remember the whole history of the game (as the type of agents' strategies suggest). Note that in the classical game theory models (von Neumann and Morgenstern, 1944) situations do correspond to states — but these are computation trees that are used there, so every state in the tree uniquely identifies a path in it as well. At the same time a concurrent game structure or an alternating transition system is based on a finite automaton that indirectly imposes a tree of possible computations. A node in the tree corresponds to a *sequence of states* in the automaton (a history).

Within original ATEL, different epistemic capabilities of agents are assumed in the context of cooperation modalities and in the context of epistemic operators. The interpretation of epistemic operators refers to the agents' capability to distinguish one *state* from another; the semantics of $\langle\langle A\rangle\rangle$ allows the agents to base their decisions upon *sequences* of states. This dichotomy reflects the way a concurrent epistemic game structure can be unraveled (Figure 4.1 in section 4.2). We believe that the dilemma whether to assign agents with the ability to remember the whole history should be made explicit in the meta-language. Therefore we will assume that relation $\sim_a$ expresses what agent a can "see" (or observe) directly from his current state (i.e. having no recall of the past except for the information that is actually "stored" in the agent's local state), and we will call it an *observational accessibility relation* to avoid confusion. The *(perfect) recall accessibility relation* for agents that do not forget can be derived from $\sim_a$ in the form of relation $\approx_a$ over histories.

As the past is important when it comes to epistemic state of agents with perfect recall, knowledge operators should be given semantics in which the past is included. Thus, formulae like $K_a\varphi$ should be interpreted over paths rather than states of the system. The new semantics we propose for ATEL* in section 4.5 (meant as a logic for agents with finite set of states and perfect recall) draws much inspiration from branching-time logics that incorporate past in their scope (Laroussinie and Schnoebelen, 1995). The simpler case — agents with bounded memory — is also interesting. We will discuss it in section 4.4, proposing a logic aimed at observational properties of agents.

4.3.4 Feasible Strategies for Groups of Agents

The dichotomy between having a strategy *de re* and *de dicto* was discussed in section 4.3.2. The first notion is arguably more important if we want to express what agents with incomplete information can really *enforce*. In order to restrict the semantics of the cooperation modalities to feasible plans only, we suggest to rule out strategies with choices that cannot be deterministically executed by the players (via redefinition of the set of strategies available to agents) and to require that a player is able to identify a winning strategy (via redefinition of function out: all the computations must be considered that are possible from the agent's perspective — and not only the objectively possible ones).

This looks relatively straightforward for a single agent: $\langle\langle a\rangle\rangle_{K(a)}\Phi$ should mean: "a has a uniform strategy to enforce Φ and he knows that if he executes the strategy then he will bring about Φ" (cf. Definition 4.1 and Example 4.4). In such a case, there is nothing that can prevent a from executing it. The situation is not so simple for a coalition of agents. The coalition should be able to identify a winning strategy — but in what way? Suppose we require that this is common knowledge among the agents that F_A is a winning strategy — would that be enough? Unfortunately, the answer is no.

Example 4.5 Consider the following variant of the *matching pennies* game. There are two agents – both with a coin – and each can choose to show heads or tails. If they choose the same, they win, otherwise they loose. There are two obvious collective strategies that result in victory for the coalition, even when we consider common knowledge *de re*; hence $\exists_{F_{\{1,2\}}} C_{\{1,2\}}[F_{\{1,2\}}]\mathtt{win}$. However, both agents have to choose *the same* winning strategy, so it is still hard for them to win this game! In fact, they cannot play it successfully with no additional communication between them. □

Thus, even common knowledge among A of a winning strategy F_A for them does not imply that the agents from A can automatically apply F_A as long as there are other winning strategies commonly identified by A. It means that the coalition must have a strategy selection criterion upon which all agents from A agree. How have they come to this agreement? Through some additional communication "outside the model"? But why should not distributed knowledge be used instead then – if the agents are allowed to communicate outside the model at all, perhaps they can share their private knowledge too? Other settings make sense as well: there can be a leader within the team that can assign the rest of the team with their strategies (then it is sufficient that the strategy is identified by the leader). Or, the leader may even stay out of the group (then he is not a member of the coalition that executes the plan). In order to capture the above intuitions in a general way, we propose to extend the simple cooperation modality $\langle\langle A\rangle\rangle$ to a family of operators: $\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\Phi$ with the intended meaning that coalition A has a (uniform) strategy to enforce Φ, and the strategy can be identified by agents $\Gamma \subseteq \mathbb{A}\mathrm{gt}$ in epistemic mode $\mathcal{K}$ (where $\mathcal{K}$ can be any of the epistemic operators K, C, E, D):

$\lambda \models \langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\Phi$ iff A have a collective uniform strategy F_A such that for every $\Lambda \in out^*_{\mathcal{K}(\Gamma)}(\lambda, F_A)$ we have that Φ holds in Λ.

These operators generalize Jonker's cooperation modalities with indices: $\langle\langle A\rangle\rangle_C$, $\langle\langle A\rangle\rangle_E$ and $\langle\langle A\rangle\rangle_{K_i}$, introduced in (Jonker, 2003).

We will use the generic notation $\approx^{\mathcal{K}}_{\Gamma}$ to denote the (path) indistinguishability relation for agents Γ in epistemic mode $\mathcal{K}$:

$$\lambda \approx^{\mathcal{K}}_{\Gamma} \lambda' \quad \text{iff} \quad \lambda[i] \sim^{\mathcal{K}}_{\Gamma} \lambda'[i] \text{ for every } i.$$

Function $out^*_{\mathcal{K}(\Gamma)}(\lambda, F_A)$ returns the computations that are possible from the viewpoint of group Γ (with respect to knowledge operator $\mathcal{K}$) after history λ took place:

$$out^*_{\mathcal{K}(\Gamma)}(\lambda, F_A) \;=\; \{\Lambda \mid \Lambda \text{ starts with } \lambda' \text{ such that } \lambda' \approx^{\mathcal{K}}_{\Gamma} \lambda, \text{ and the rest of } \Lambda \text{ is consistent with } F_A\}$$

Examples include:

- $\langle\langle A\rangle\rangle_{C(A)}\Phi$: the agents from A have a collective strategy to enforce Φ and the strategy is common knowledge in A. This requires the least amount of additional communication. It is in fact sufficient that the agents from A agree upon some total order over their group strategies at the beginning of the game (the lexicographical order, for instance) and that they will always choose the maximal winning strategy with respect to this order;

- $\langle\langle A\rangle\rangle_{E(A)}\Phi$: coalition A has a collective strategy to enforce Φ and everybody in A knows that the strategy is winning;

- $\langle\langle A\rangle\rangle_{D(A)}\Phi$: the agents from A have a strategy to enforce Φ and if they share their knowledge at the current state, they can identify the strategy as winning;

- $\langle\langle A\rangle\rangle_{K(a)}\Phi$: the agents from A have a strategy to enforce Φ, and a can identify the strategy and give them orders how to achieve the goal;

- $\langle\langle A\rangle\rangle_{D(\Gamma)}\Phi$: group Γ acts as a kind of "headquarters committee": they can fully cooperate within Γ (at the current state) to find a strategy to achieve Φ. The strategy is aimed for A, so it must be uniform for agents from A.

Note also that $\langle\langle A\rangle\rangle_{C(\varnothing)}\Phi$ means that A have a uniform strategy to achieve Φ (but they may be unaware of it, and of the strategy itself), because $\sim^{C}_{\varnothing}$ is the accessibility relation when complete information is available. In consequence, $\mathcal{K}_A\langle\langle A\rangle\rangle_{C(\varnothing)}\Phi$ captures the notion of having a strategy *de dicto* from section 4.3.2. Since the original ATL meaning of $\langle\langle A\rangle\rangle\Phi$ (there is a *complete information* strategy to accomplish Φ) does not seem to be expressible with the new modalities, we suggest to leave the operator in the language as well.

Example 4.6 Let us consider the modified variable client/server system from Figure 3.3 once more to show how the new modalities work:

- $x = 1 \rightarrow \langle\langle s\rangle\rangle_{K(s)}\bigcirc x = 1$, because every time s is in q_1, he can choose to reject the client's request (and he knows it, because he can distinguish q_1 from the other states);

- $\neg x = 2 \rightarrow \neg\langle\langle s, c\rangle\rangle_{K(c)}\bigcirc x = 2$, because – for every history $(qq'q''...q_1)$ – c cannot distinguish it from $(qq'q''...q_0)$ and vice versa, so he cannot effectively identify a uniform strategy;

- $x = 2 \rightarrow \neg\langle\langle s\rangle\rangle_{K(s)}\bigcirc x = 2 \wedge \neg\langle\langle c\rangle\rangle_{K(c)}\bigcirc x = 2$, because c has no action to request no change, and s is unable to identify the current state;

- however, $x = 2 \rightarrow \langle\langle s\rangle\rangle_{K(c)}\bigcirc x = 2$. The client can "indicate" the right strategy to the server;

- $x = 0 \rightarrow \neg\langle\langle s\rangle\rangle_{K(s)}\bigcirc x = 0 \wedge \neg\langle\langle s\rangle\rangle_{K(c)}\bigcirc x = 0 \wedge \langle\langle s\rangle\rangle_{D(\{s,c\})}\bigcirc x = 2$: only if s and c join their pieces of knowledge, they can identify a feasible strategy for s;
- $x = 1 \rightarrow \langle\langle s, c\rangle\rangle_{E(\{c,s\})}\bigcirc\neg x = 0 \wedge \neg\langle\langle c, s\rangle\rangle_{C(\{s,c\})}\bigcirc\neg x = 0$: both processes can identify a collective strategy to change x from 1 to 0, but they are not sure if the other party can identify it too.

□

The next two sections follow with a formalization of the intuitions described so far.

4.4 ATOL: a Logic of Observations

Assigning an agent the ability to remember everything that has happened in the past seems unrealistic in many cases. Both humans and software agents have obviously limited memory capabilities. On the other hand, we usually cannot know precisely what the agents in question will actually remember from their history – in such situations perfect recall can be attractive as the upper bound approximation of the agents' potential. Some agents may also enhance their capacity (install new memory chips when more storage space is needed, for instance). In this case the memory of the agents is finite, but not bounded, and they cannot be appropriately modeled with bounded recall apparatus.

We believe that both settings are interesting and worth further investigation. In this section, we start with introducing the simpler case of imperfect recall in the form of Alternating-time Temporal Observational Logic (ATOL). As the original ATL and ATEL operators $\langle\langle A\rangle\rangle$ were defined to describe agents with perfect recall, it seems best to leave them with this meaning. Instead, we will use a new modality $\langle\langle A\rangle\rangle^{\bullet}$ to express that the agents from A can enforce a property while their ability to remember is bounded. When uniform strategies are to be considered, the operator will be used with an appropriate subscript in the way proposed in Sections 4.3.2 and 4.3.4.

If agents are assumed to remember no more than n most recent positions in a finite automaton, a new automaton can be proposed in which the last n positions are included in the states and the epistemic links define what the agents actually remember in every situation. Thus, for every model in which the agents can remember a limited number of past events, an equivalent model can be constructed in which they can recall no past at all (cf. Example 4.8). ATOL is a logic for agents with no recall – it refers to the features that agents can *observe* on the spot. Note, however, that these are observations in the broadest sense, including perceptions of the external world, and the internal (local) state of the agent.

4.4.1 Syntax

An ATOL formula is one of the following:

- p, where p is an atomic proposition;

- $\neg\varphi$ or $\varphi \vee \psi$, where φ, ψ are ATOL formulae;
- $Obs_a\varphi$, where a is an agent and φ is a formula of ATOL;
- $CO_A\varphi$, $EO_A\varphi$ and $DO_A\varphi$, where A is a set of agents and φ is a formula of ATOL;
- $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\bigcirc\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\Box\varphi$, or $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\varphi\mathcal{U}\psi$, where φ, ψ are ATOL formulae and A is a set of agents, and γ an agent (not necessarily a member of A).
- $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\bigcirc\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\Box\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\varphi\mathcal{U}\psi$, where φ, ψ are ATOL formulae and A and Γ are sets of agents and $\Theta(\Gamma) \in \{CO(\Gamma), DO(\Gamma), EO(\Gamma)\}$.

Formula $Obs_a\varphi$ reads: "agent a observes that φ". Operators CO_A, EO_A and DO_A refer to "common observation", "everybody sees" and "distributed observation" modalities. The informal meaning of $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\Phi$ is:

> "group A has a strategy to enforce Φ, and agent γ can see the strategy."

The common sense reading of $\langle\langle A\rangle\rangle^{\bullet}_{CO(\Gamma)}\Phi$ is that coalition A has a collective strategy to enforce Φ, and the strategy itself is a common observation for group Γ. The meaning of $\langle\langle A\rangle\rangle^{\bullet}_{EO(\Gamma)}\Phi$ and $\langle\langle A\rangle\rangle^{\bullet}_{DO(\Gamma)}\Phi$ is analogous. Since the agents are assumed to have no recall in ATOL, the choices they make within their strategies must be based on the current state only. As we want them to specify deterministic plans under incomplete information, the plans should be uniform strategies as well.

Note that ATOL contains only formulae for which the past is irrelevant and no specific future branch is referred to, so it is sufficient to evaluate them over single states of the system.

4.4.2 Semantics

Formulae of Alternating-time Temporal Observational Logic are interpreted in *concurrent observational game structures*:

$$S = \langle k, Q, \Pi, \pi, \sim_1, ..., \sim_k, d, o\rangle$$

in which agents have the same choices in indistinguishable states: for every q, q' such that $q \sim_a q'$ it is required that $d_a(q) = d_a(q')$. To specify plans, they can use uniform strategies with no recall.

Definition 4.2 *A* uniform strategy with no recall *is a function* $v_a : Q \to \mathbb{N}$ *for which:*

- $v_a(q) \leq d_a(q)$ *(the strategy specifies valid decisions),*
- *if two states are indistinguishable* $q \sim_a q'$ *then* $v_a(q) = v_a(q')$.

As usual, a collective strategy V_A assigns every agent $a \in A$ with one strategy v_a. The group observational accessibility relations can also be defined in the standard way:

$$\begin{array}{lcl} \sim_A^{DO} & = & \bigcap_{a \in A} \sim_a; \\ \sim_A^{EO} & = & \bigcup_{a \in A} \sim_a; \\ \sim_A^{CO} & \text{is} & \text{the reflexive and transitive closure of } \sim_A^{EO}. \end{array}$$

The set of computations that are possible from agent γ's point of view, consistent with strategy V_A and starting from state q, can be defined as:

$$out_{Obs(\gamma)}(q, V_A) = \{\Lambda | \ \Lambda \text{ is consistent with } V_A \text{ and } \Lambda[0] \sim_\gamma q\}.$$

Definition 4.3 *More generally, for* $\Gamma \subseteq \mathbb{A}\text{gt}$*, and* Θ *being any of the collective observation modes CO, EO, DO:*

$$out_{\Theta(\Gamma)}(q, V_A) = \{\Lambda | \ \Lambda \textit{ is consistent with } V_A \textit{ and } \Lambda[0] \sim_\Gamma^\Theta q\},$$

Definition 4.4 *We define the semantics of* ATOL *with the following rules:*

$q \models p$	iff	$p \in \pi(q)$
$q \models \neg\varphi$	iff	$q \not\models \varphi$
$q \models \varphi \vee \psi$	iff	$q \models \varphi$ or $q \models \psi$
$q \models Obs_a\ \varphi$	iff	for every $q' \sim_a q$ we have $q' \models \varphi$
$q \models \langle\langle A \rangle\rangle^\bullet_{Obs(\gamma)} \bigcirc \varphi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{Obs(\gamma)}(q, V_A)$ we have $\Lambda[1] \models \varphi$
$q \models \langle\langle A \rangle\rangle^\bullet_{Obs(\gamma)} \Box \varphi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{Obs(\gamma)}(q, V_A)$ we have $\Lambda[i] \models \varphi$ for all $i = 0, 1, \ldots$
$q \models \langle\langle A \rangle\rangle^\bullet_{Obs(\gamma)} \varphi \mathcal{U} \psi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{Obs(\gamma)}(q, V_A)$ there is a $k \geq 0$ such that $\Lambda[k] \models \psi$ and $\Lambda[i] \models \varphi$ for all $0 \leq i \leq k$
$q \models \Theta_A\ \varphi$	iff	for every $q' \sim_A^\Theta q$ we have $q' \models \varphi$
$q \models \langle\langle A \rangle\rangle^\bullet_{\Theta(\Gamma)} \bigcirc \varphi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{\Theta(\Gamma)}(q, V_A)$ we have $\Lambda[1] \models \varphi$
$q \models \langle\langle A \rangle\rangle^\bullet_{\Theta(\Gamma)} \Box \varphi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{\Theta(\Gamma)}(q, V_A)$ we have $\Lambda[i] \models \varphi$ for all $i = 0, 1, \ldots$
$q \models \langle\langle A \rangle\rangle^\bullet_{\Theta(\Gamma)} \varphi \mathcal{U} \psi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{\Theta(\Gamma)}(q, V_A)$ there is a $k \geq 0$ such that $\Lambda[k] \models \psi$ and $\Lambda[i] \models \varphi$ for all $0 \leq i \leq k$

Remark 4.1 *Note that operators* Obs_a *and* $\langle\langle A \rangle\rangle^\bullet_{Obs(\gamma)}$ *are in fact redundant:*

- $Obs_a\varphi \equiv CO_{\{a\}}\varphi$;
- $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\bigcirc\varphi \equiv \langle\langle A\rangle\rangle^{\bullet}_{CO(\{\gamma\})}\bigcirc\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\Box\varphi \equiv \langle\langle A\rangle\rangle^{\bullet}_{CO(\{\gamma\})}\Box\varphi$, *and* $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\varphi\mathcal{U}\psi \equiv \langle\langle A\rangle\rangle^{\bullet}_{CO(\{\gamma\})}\varphi\mathcal{U}\psi$.

ATOL generalizes ATL_{ir} from (Schobbens, 2003). Also, $\langle\langle A\rangle\rangle^{\bullet}\bigcirc\varphi$, $\langle\langle A\rangle\rangle^{\bullet}\Box\varphi$ and $\langle\langle A\rangle\rangle^{\bullet}\varphi\mathcal{U}\psi$ can be added to ATOL for complete information (i.e. possibly non-uniform) strategies with no recall – corresponding to the ATL_{Ir} logic (Schobbens, 2003).

Proposition 4.2 *Model checking* ATOL *is NP-complete.*

Proof:

1. ATOL model checking is NP-easy: the only difference between ATOL and the preliminary version of ATEL from (van der Hoek and Wooldridge, 2002) are the type of a strategy and the set of feasible computations $out(q, F_A)$ in the semantics of cooperation modalities. Note that for every $\langle\langle A\rangle\rangle_{\Theta(\Gamma)}$ the number of available strategies with no recall is finite, so the agents can guess the strategy nondeterministically. The algorithm produces then the set of states $Q_1 \subseteq Q$ for which A have a uniform strategy to achieve Φ (being possibly unable to identify the strategy). We can now obtain Q_2 for which $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\Phi$ by guessing a subset of Q_1 where all the states $\Theta(\Gamma)$-accessible from Q_2 are in Q_1.

2. ATOL model checking is NP-hard: ATOL subsumes ATL_{ir} from (Schobbens, 2003), for which the problem is already NP-complete.

□

Thus, ATOL model checking is intractable (unless P=NP). We already pointed out that model checking of ATOL formulae can be approximated via checking their ATEL counterparts in all the indistinguishable states (cf. Section 3.4.5 and especially Remark 3.8); the idea resembles closely the algorithm of Monte Carlo minimaxing (Corlett and Todd, 1985; Ginsberg, 1999). Using it, we often get suboptimal results, but the process has polynomial complexity again.

Remark 4.3 ATOL *(syntactically) subsumes most of* CTL. *Although none of* $\langle\langle \mathbb{A}\text{gt}\rangle\rangle^{\bullet}_{\Theta(\Gamma)}$ *is equivalent to the* CTL*'s* E, *yet still the universal path quantifier* A *can be expressed with* $\langle\langle\varnothing\rangle\rangle^{\bullet}_{CO(\varnothing)}$. *Thus also most of "there is a path" formulae can also be redefined:*

$$\begin{aligned}\mathsf{E}\bigcirc\varphi &\equiv \neg\langle\langle\varnothing\rangle\rangle^{\bullet}_{CO(\varnothing)}\bigcirc\neg\varphi,\\ \mathsf{E}\Box\varphi &\equiv \neg\langle\langle\varnothing\rangle\rangle^{\bullet}_{CO(\varnothing)}\Diamond\neg\varphi,\\ \mathsf{E}\Diamond\varphi &\equiv \neg\langle\langle\varnothing\rangle\rangle^{\bullet}_{CO(\varnothing)}\Box\neg\varphi.\end{aligned}$$

Remark 4.4 ATOL *(without the perfect information modalities* $\langle\langle A\rangle\rangle^{\bullet}$*) does* not *cover the expressive power of full* CTL. *Unlike in* ATL *(and even* ATL_{Ir}*),* $\mathsf{E}\varphi\mathcal{U}\psi$ *cannot be translated to* $\langle\langle \mathbb{A}\text{gt}\rangle\rangle^{\bullet}_{CO(\varnothing)}\varphi\mathcal{U}\psi$. *Moreover,* $\mathsf{E}\varphi\mathcal{U}\psi$ *cannot be expressed as a combination of* $\mathsf{A}\varphi\mathcal{U}\psi$, $\mathsf{E}\Diamond\varphi$, $\mathsf{E}\Box\varphi$, $\mathsf{A}\Box\varphi$, $\mathsf{E}\bigcirc\varphi$, *and* $\mathsf{A}\bigcirc\varphi$ *(cf. (Laroussinie, 1995)).*

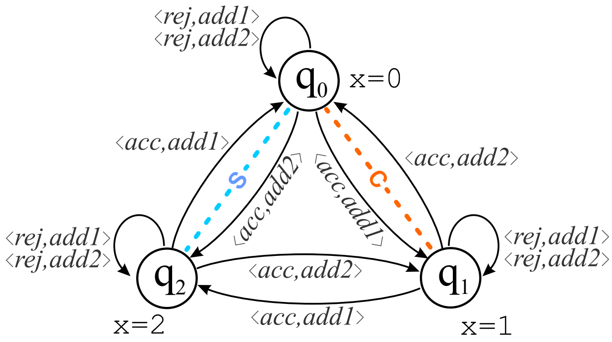


Figure 4.4: The controller/client problem again

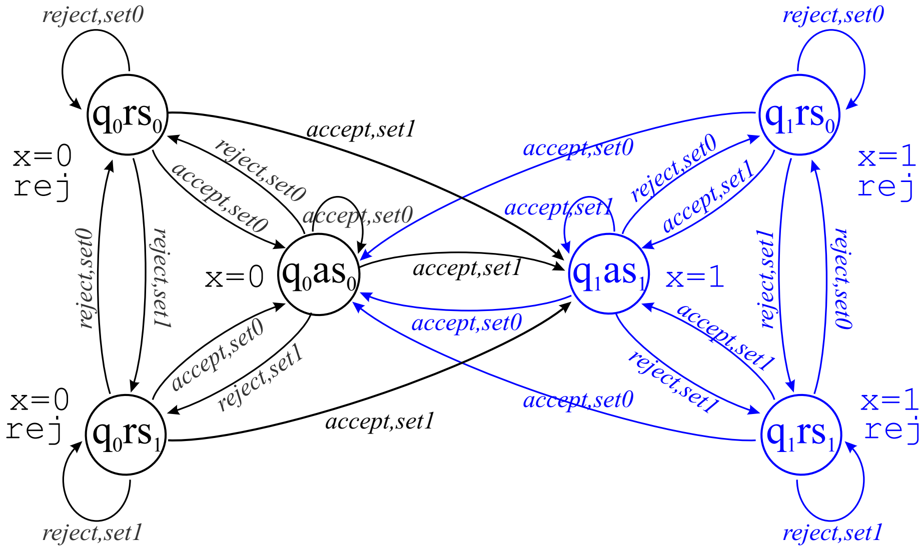
Remark 4.5 *Note that* ATL$_{Ir}$ *is equivalent to* ATL *(Schobbens, 2003), so* ATOL *begins to cover the expressive power of* CTL *as soon as we add the perfect information modalities to* ATOL.

4.4.3 Examples

Let us consider a few examples to see how properties of agents and their coalitions can be expressed with ATOL. We believe that especially Example 4.9 demonstrates the potential of ATEL in reasoning about limitations of agents, and the ways they can be overcome.

Example 4.7 First, we can have a look at the variable client/server system from Example 3.3 again, this time in the form of a concurrent observational game structure (see Figure 4.4). Note how the observational relation is defined: if we think of x in binary representation x_1x_2, we have that c can observe x_1, whereas s observers x_2. The following formulae are valid in the system:

- $Obs_s x = 1 \vee Obs_s \neg x = 1$: the server can recognize whether the value of x is 1 or not;
- $\langle\langle s, c\rangle\rangle^{\bullet}_{CO(s,c)} \bigcirc \neg x = 2$: the agents have a strategy *de re* to avoid $x = 2$ in the next step. For instance, the client can always execute *add1*, and the server rejects the request in q_1 and accepts otherwise;
- $x = 2 \rightarrow \neg\langle\langle s\rangle\rangle^{\bullet}_{Obs(s)} \bigcirc (x = 2) \wedge \langle\langle s\rangle\rangle^{\bullet}_{Obs(c)} \bigcirc (x = 2)$: The server s must be hinted a strategy by c if he wants the variable to retain the value of 2. To see why this is true, suppose that $x = 2$. We have to find a strategy v_s such that for every $\Lambda \in out_{Obs(c)}(q_2, v_s)$, we have $\Lambda[1] \models \bigcirc(x = 2)$. Let v_s be the strategy that picks *rej* in all states. Then, obviously, v_s is an incomplete information strategy. All the computation paths consistent with this strategy are $q_0q_0\ldots$, $q_1q_1\ldots$ and $q_2q_2\ldots$. The runs from those that are in $out_{Obs(c)}(q_2, v_s)$ are those that start in q_2, so the only element we retain is $\Lambda = q_2q_2\ldots$. Obviously, for this Λ, we have $\Lambda[1] \models (x = 2)$. To further see that in q_2 we have $\neg\langle\langle s\rangle\rangle^{\bullet}_{Obs(s)} \bigcirc (x = 2)$,



$q\alpha\beta \sim_s q'\alpha'\beta'$
iff $q = q'$ and $\alpha = \alpha'$

$q_1q_2 \sim_c q_1'q_2'$
iff $q = q'$ and $\beta = \beta'$

Figure 4.5: Agents with some memory of the past. Proposition `rej` holds in the states immediately after a request has been rejected.

assume that there is some strategy v_s such that for every $\Lambda \in out_{Obs(c)}(q_2, v_s)$ we have $\Lambda[1] \models (x = 2)$. The only strategy v_s that works here chooses *rej* in q_2. Since v_s has to be an incomplete information strategy, v_s prescribes *rej* in q_2 as well. But the runs generated by this v_s in $out_{Obs(c)}(q_2, v_s)$ are $\Lambda = q_2q_2\ldots$ and $\Lambda' = q_0q_0\ldots$. Obviously, we do not have $\Lambda'[1] \models x = 2$;

- $\langle\langle s, c\rangle\rangle^{\bullet}_{CO(s,c)}\Diamond(Obs_s x = 0 \vee Obs_s x = 1 \vee Obs_s x = 2) \wedge \langle\langle s, c\rangle\rangle^{\bullet}_{CO(s,c)}\Diamond(Obs_c x = 0 \vee Obs_c x = 1 \vee Obs_c x = 2) \wedge \neg\langle\langle s, c\rangle\rangle^{\bullet}_{CO(s,c)}\Diamond(EO_{\{s,c\}} x = 0 \vee EO_{\{s,c\}} x = 1 \vee EO_{\{s,c\}} x = 2)$: the agents have a way to make the value of x observable for any of them, but they have no strategy to make it observable to both of them at the same moment.

□

Example 4.8 Let us go back to the first variable/controller system with only two states (Example 2.3). The system can be modified to include bounded memory of the players: for instance, we may assume that each agent remembers the last decision he made. The resulting concurrent observational game structure is shown in Figure 4.5. For this structure, we may for instance demonstrate that:

- s can always reject the claim: $\mathsf{A}\Box\langle\langle s\rangle\rangle^{\bullet}_{Obs(s)}\bigcirc rej$, where $\mathsf{A} \equiv \langle\langle\varnothing\rangle\rangle^{\bullet}_{CO(\varnothing)}$ (cf. Remark 4.3);

- if s rejects the claims then the value of x will not change – and s can see it: $Obs_s[(x = 0 \rightarrow \mathsf{A}\bigcirc(rej \rightarrow x = 0)) \wedge (x = 1 \rightarrow \mathsf{A}\bigcirc(rej \rightarrow x = 1)) \wedge (x = 2 \rightarrow \mathsf{A}\bigcirc(rej \rightarrow x = 2))]$. Note that this kind of formulae can be used in ATOL to specify results of particular strategies in the object language (in this case: the "always reject" strategy).

□

Example 4.9 Let us consider a train controller example similar to the one from (Alur et al., 2000; van der Hoek and Wooldridge, 2002). There are two trains tr_1, tr_2, and a controller c that can let them into the tunnel. The algorithm of train tr_i is sketched in Figure 4.6. Each train can opt to stay out of the tunnel (action s) for some time – its local state is "away" (a_i) then. When the train wants to enter the tunnel (e), it must wait (state w_i) until the controller lights a green light for it (action let_i from the controller). In the tunnel (t_i), the train can again decide to stay for some time (s) or to exit (e). There is enough vocabulary to talk about the position of each train (propositions $\mathsf{a1}, \mathsf{w1}, \mathsf{t1}, \mathsf{a2}, \mathsf{w2}$ and $\mathsf{t2}$).

The set of possible situations (global states) is

$$Q = \{a_1a_2, a_1w_2, a_1t_2, w_1a_2, w_1w_2, w_1t_2, t_1a_2, t_1w_2, t_1t_2\}.$$

The transition function for the whole system, and the accessibility relations are depicted in Figure 4.7. Every train can observe only its own position. The controller is not very capable observationally: it can see which train is away – but nothing more. When one of the trains is away and the other is not, c has to light the green light for the latter.[2] The trains crash if they are in the tunnel at the same moment ($crash \equiv t1 \wedge t2$), so the controller should not let a train into the tunnel if the other train is inside. Unfortunately:

- c is not able to do so: $\neg\langle\langle c\rangle\rangle^{\bullet}_{Obs(c)}\Box\neg crash$, because it has to choose the same option in w_1t_2 and w_2t_1. Note that the controller would be able to keep the trains from crashing if it had perfect information: $\neg\langle\langle c\rangle\rangle^{\bullet}\Box\neg crash$, which shows exactly that insufficient epistemic capability of c is the source of this failure;

- on the other hand, a train (say, tr_1) can hint the right strategy (pass a signal) to the controller every time it is in the tunnel, so that there is no crash in the next moment: $\mathsf{A}\Box(t1 \rightarrow \langle\langle c\rangle\rangle^{\bullet}_{Obs(tr_1)}\bigcirc\neg crash)$;

- when tr_1 is out of the tunnel, then c can choose the strategy of letting tr_2 in if tr_2 is not away (and choosing let_1 else) to succeed in the next step: $\mathsf{A}\Box(\neg t1 \rightarrow \langle\langle c\rangle\rangle^{\bullet}_{Obs(c)}\bigcirc\neg crash)$;

- two last properties imply also that $\mathsf{A}\Box\langle\langle c\rangle\rangle^{\bullet}_{DO(c,tr_1)}\bigcirc\neg crash$: the controller can avoid the crash when he has enough communication from tr_1;

[2] This is meant to impose fair access of the trains to the tunnel: note that when tr_i wants to enter the tunnel, it must be eventually allowed if only the other train does not stay in the tunnel for ever. Adding explicit fairness conditions, like in Fair ATL (Alur et al., 2002), would probably be a more elegant solution, but it goes far beyond the scope of the example and the chapter.

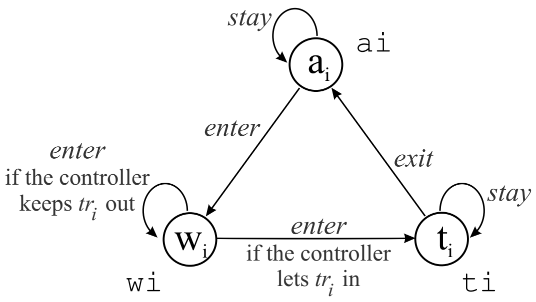


Figure 4.6: Train template: tr_i for the train controller problem

	s,s,let_1	s,s,let_2	s,e,let_1	s,e,let_2	e,s,let_1	e,s,let_2	e,e,let_1	e,e,let_2
a_1a_2	a_1a_2	a_1a_2	a_1w_2	a_1w_2	w_1a_2	w_1a_2	w_1w_2	w_1w_2
a_1w_2	–	–	–	a_1t_2	–	–	–	w_1t_2
w_1a_2	–	–	–	–	t_1a_2	–	t_1w_2	–
w_1w_2	–	–	–	–	–	–	t_1w_2	w_1t_2
a_1t_2	–	a_1t_2	–	a_1a_2	–	w_1t_2	–	w_1a_2
t_1a_2	t_1a_2	–	t_1w_2	–	a_1a_2	–	a_1w_2	–
w_1t_2	–	–	–	–	t_1t_2	w_1t_2	t_1a_2	w_1a_2
t_1w_2	–	–	t_1w_2	t_1t_2	–	–	a_1w_2	a_1t_2
t_1t_2	t_1t_2	t_1t_2	t_1t_2	t_1t_2	t_1t_2	t_1t_2	t_1t_2	t_1t_2

$$q_1q_2 \sim_{tr_1} q'_1q'_2 \quad \text{iff} \quad q_1 = q'_1$$
$$q_1q_2 \sim_{tr_2} q'_1q'_2 \quad \text{iff} \quad q_2 = q'_2$$

$\sim_c$	a_1a_2	a_1w_2	w_1a_2	w_1w_2	a_1t_2	t_1a_2	w_1t_2	t_1w_2	t_1t_2
a_1a_2	+								
a_1w_2		+			+				
w_1a_2			+			+			
w_1w_2				+			+	+	+
a_1t_2		+			+				
t_1a_2			+			+			
w_1t_2				+			+	+	+
t_1w_2				+			+	+	+
t_1t_2				+			+	+	+

Figure 4.7: Transitions and observational accessibility for the system with two trains and a controller

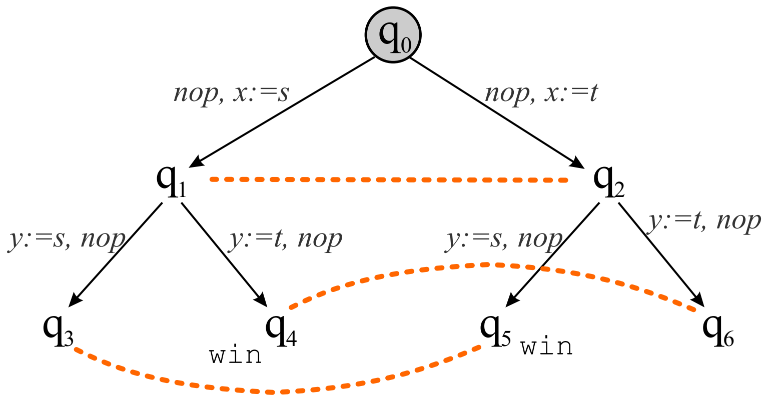


Figure 4.8: A model for an *IF* game: $S[\forall x \exists y/x \; x \neq y]$. To help the reader, the nodes in which Falsifier makes a move are marked with grey circles; it is Verifier's turn at all the other nodes.

- however, $\neg\langle\!\langle c \rangle\!\rangle^{\bullet}_{DO(c,tr_1)} \Box \neg crash$, so a one-time communication is not enough;
- finally, c is not a very good controller for one more reason – it cannot detect a crash even if it occurs: $crash \rightarrow \neg Obs_c\, crash$.

□

Example 4.10 The last example refers to *IF* games, introduced by Hintikka and Sandu in (Hintikka and Sandu, 1997), and investigated further in (van Benthem, 2002) from a game-theoretic perspective. The metaphor of mathematical proof as a game between Verifier V (who wants to show that the formula in question is true) and Falsifier F (who wants to demonstrate the opposite) is the starting point here. One agent takes turn at each quantifier: at $\exists x$, Verifier is free to assign x with any domain object he likes, while at $\forall x$ the value is chosen by Falsifier. *IF* games generalize the idea with their "slash notation": $\exists x/y$ means that V can choose a value for x, but at the same time he must forget everything he knew about the value of y (for ever). (van Benthem, 2002) suggests that such logic games can be given a proper game-theoretical treatment too, and uses dynamic-epistemic logic to reason about the players' knowledge, their powers etc. Obviously, ATOL can be used for the same purpose.

Let us consider two *IF* games from (van Benthem, 2002): one for $\forall x \exists y/x \; x \neq y$, the other for $\forall x \exists z \exists y/x \; x \neq y$. The game trees for both games are shown in Figures 4.8 and 4.9. The arcs are labeled with j_V, j_F where j_V is the action of Verifier and j_F is the Falsifier's decision; *nop* stands for "no-operation" or "do-nothing" action. Dotted lines display V's observational accessibility links. F has perfect information in both games. It is assumed that the domain contains two objects: s and t. Atom win indicates the states in which the Verifier wins, i.e. the states in which he has been able to prove the formula in question.

We will use the trees as concurrent observational game structures to demonstrate interesting properties of the players with ATOL formulae.

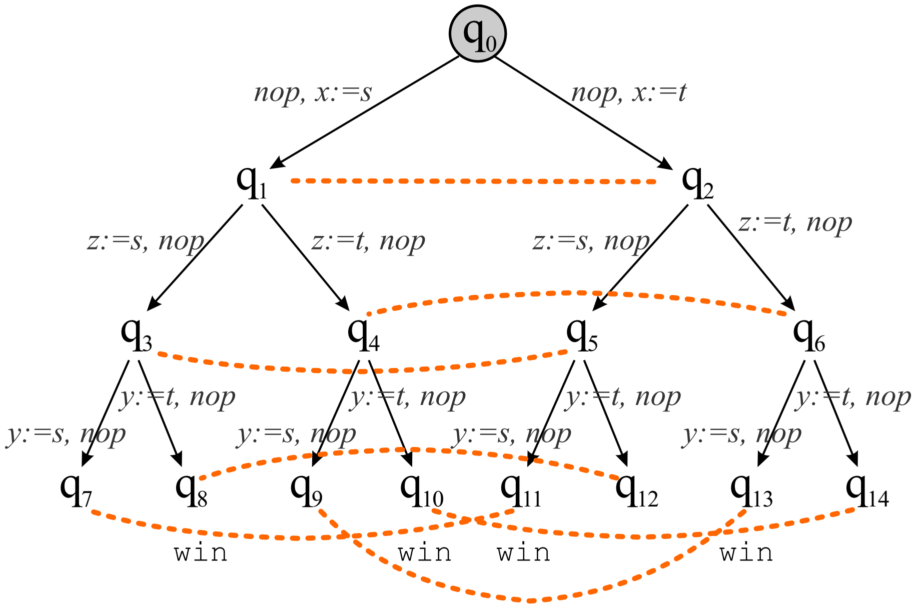


Figure 4.9: Another *IF* game: $S[\forall x \exists z \exists y/x\; x \neq y]$.

- $S[\forall x \exists y/x\; x \neq y], q_0 \models \neg\langle\langle V\rangle\rangle^{\bullet}_{Obs(V)}\Diamond win$: Verifier has no uniform strategy to win this game;
- note that $S[\forall x \exists y/x\; x \neq y], q_0 \models \neg\langle\langle F\rangle\rangle^{\bullet}_{Obs(F)}\Box\neg win$: Falsifier has no power to prevent V from winning as well in the first game – in other words, the game is non-determined. Thus, the reason for V's failure lies in his insufficient epistemic abilities – in the second move, to be more specific: $S[\forall x \exists y/x\; x \neq y], q_0 \models \langle\langle V\rangle\rangle^{\bullet}_{Obs(V)}\bigcirc\langle\langle V\rangle\rangle^{\bullet}_{CO(\varnothing)}\Diamond win$;
- the vacuous quantifier in (B) does matter a lot: V can use it to store the actual value of x, so $S[\forall x \exists z \exists y/x\; x \neq y], q_0 \models \langle\langle V\rangle\rangle^{\bullet}_{Obs(V)}\Diamond win$.
- Verifier has a strategy that guarantees win (see above), but he will never be able to observe that he has actually won:

$$S[\forall x \exists z \exists y/x\; x \neq y], q_0 \models \neg\langle\langle V\rangle\rangle^{\bullet}_{Obs(V)}\Diamond Obs_V win.$$

□

Giving a complete axiomatization for ATOL is beyond the scope of this chapter. We only mention a few tautologies below.

Proposition 4.6 *The following are valid* ATOL *properties:*

1. $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\Phi \rightarrow Obs_\gamma\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\Phi$*: if* γ *is able to identify* A*'s strategy to bring about* Φ*, then he observes that* A *have such a strategy, too;*

2. *more generally:* $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\Phi \rightarrow \Theta_\Gamma\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\Phi$*;*

3. $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\Phi \rightarrow \Theta_\Gamma\langle\langle A\rangle\rangle^{\bullet}_{CO(\varnothing)}\Phi$*: if A have a strategy* de re *in any sense, then they have also a strategy* de dicto *in the same sense;*

4. *having a uniform strategy implies having a complete information strategy:* $\langle\langle A\rangle\rangle^{\bullet}_{CO(\varnothing)}\Phi \rightarrow \langle\langle A\rangle\rangle^{\bullet}\Phi$.

4.5 ATEL-R*: Knowledge and Time with no Restraint

Real agents have finite memory and unless they can extend their capacity when necessary (hence making the memory finite, but unbounded), models with no recall can be used for them. However, even if we know that an agent has limited memory capabilities, we seldom know which observations he will actually decide to remember. Models with no recall exist for many problems, but they are often extremely large and must be constructed on the fly for every particular setting. Assigning agents with perfect recall can be a neat way to get rid of these inconveniences, although at the expense of making the agents remember (and accomplish) too much. Our language to talk about agents with recall – Alternating-time Temporal Epistemic Logic with Recall (ATEL-R*) – includes the following formulae:

- p, where p is an atomic proposition;
- $\neg\varphi$ or $\varphi \vee \psi$, where φ, ψ are ATEL-R* formulae;
- $\bigcirc\varphi$ or $\varphi\mathcal{U}\psi$, where φ, ψ are ATEL-R* formulae.
- $\mathcal{K}_A\ \varphi$, where $\mathcal{K}$ is any of the collective knowledge operators: C, E, D, A is a set of agents, and φ is an ATEL-R* formula;
- $\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\ \varphi$, where A, Γ are sets of agents, $\mathcal{K} = C, E, D$, and φ is an ATEL-R* formula.

We would like to embed the observational logic ATOL, and modalities for strategies with complete information into ATEL-R* in a general way. Past time operators can be also useful in the context of perfect recall, so the following formulae are meant to be a part of ATEL-R* as well ($\Theta = CO, EO, DO$ and $\mathcal{K} = C, E, D$):

- $\Theta_A\ \varphi$;
- $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{\mathcal{K}(\Gamma)}\varphi$, $\langle\langle A\rangle\rangle_{\Theta(\Gamma)}\varphi$;
- $\langle\langle A\rangle\rangle^{\bullet}\varphi$, $\langle\langle A\rangle\rangle\varphi$;
- $\bigcirc^{-1}\varphi$ ("previously φ") and $\varphi\,\mathcal{S}\,\psi$ ("φ since ψ") .

Several derived operators can be defined:

- $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$ etc.;

- $K_a\varphi \equiv C_{\{a\}}\varphi$ and $\langle\langle A\rangle\rangle_{K(\gamma)}\Phi \equiv \langle\langle A\rangle\rangle_{C(\{\gamma\})}\varphi$;
- $Obs_a\varphi \equiv CO_{\{a\}}\varphi$ and $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\Phi \equiv \langle\langle A\rangle\rangle^{\bullet}_{CO(\{\gamma\})}\Phi$;
- $\Diamond\varphi \equiv true\,\mathcal{U}\varphi$ and $\Box\varphi \equiv \neg\Diamond\neg\varphi$;
- $\Diamond^{-1}\varphi \equiv true\,\mathcal{S}\,\varphi$ and $\Box^{-1}\varphi \equiv \neg\Diamond^{-1}\neg\varphi$;
- $\mathsf{A}\varphi \equiv \langle\langle\varnothing\rangle\rangle_{C(\varnothing)}\varphi$ and $\mathsf{E}\varphi \equiv \neg\langle\langle\varnothing\rangle\rangle_{C(\varnothing)}\neg\varphi$.

4.5.1 Semantics for ATEL-R*

A few semantics have been proposed for CTL* with past time (Hafer and Thomas, 1987; Laroussinie and Schnoebelen, 1995). The semantics we use for ATEL-R* is based on (Laroussinie and Schnoebelen, 1995), where cumulative linear past is assumed: the history of the current situation increases with time and is never forgotten. In a similar way, we do not make the usual (unnecessary) distinction between state and path formulae here.

The knowledge accessibility relation for agent a is defined as before: $\lambda \approx^{\mathcal{K}}_A \lambda'$ iff $\lambda[i] \sim^{\mathcal{K}}_A \lambda'[i]$ for all i. Again, $\xi[i]$, $\xi_{|i}$, and ξ^i denote the ith position, first $i+1$ positions, and the ith suffix of ξ respectively. The semantics for ATEL-R*, proposed below, exploits also function $out^*_{\mathcal{K}(\Gamma)}(\lambda, F_A)$ which returns the set of computations that are possible from the viewpoint of group Γ (with respect to knowledge operator $\mathcal{K}$) in situation λ (i.e. after history λ took place):

$$out^*_{\mathcal{K}(\Gamma)}(\lambda, F_A) = \{\Lambda \mid \Lambda_{|n} \approx^{\mathcal{K}}_{\Gamma} \lambda \text{ and } \Lambda^n \text{ is consistent with } F_A\text{, where } n \text{ is the length of } \lambda\}.$$

Definition 4.5 *The semantics of* ATEL-R* *is defined with the following rules:*

$\Lambda, n \models p$	iff	$p \in \pi(\Lambda[n])$
$\Lambda, n \models \neg\varphi$	iff	$\Lambda, n \not\models \varphi$
$\Lambda, n \models \varphi \vee \psi$	iff	$\Lambda, n \models \varphi$ or $\Lambda, n \models \psi$
$\Lambda, n \models \bigcirc\varphi$	iff	$\Lambda, n+1 \models \varphi$
$\Lambda, n \models \varphi\mathcal{U}\psi$	iff	there is a $k \geq n$ such that $\Lambda, k \models \psi$ and $\Lambda, i \models \varphi$ for all $n \leq i < k$
$\Lambda, n \models \mathcal{K}_A\,\varphi$	iff	for every Λ' such that $\Lambda'_{|n} \approx^{\mathcal{K}}_A \Lambda_{|n}$ we have $\Lambda', n \models \varphi$ (where $\mathcal{K}$ can be any of the collective knowledge operators: C, E, D)
$\Lambda, n \models \langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\,\varphi$	iff	there exists a collective uniform strategy F_A such that for every $\Lambda' \in out^*_{\mathcal{K}(\Gamma)}(\Lambda_{|n}, F_A)$ we have $\Lambda', n \models \varphi$.

We believe that adding past time operators to ATEL-R* does not change its expressive power – the same way as CTL*+Past has been proven equivalent to CTL* (Hafer and Thomas, 1987; Laroussinie and Schnoebelen, 1995). However, explicit past tense constructs in the language enable expressing history-oriented properties in a natural and easy way.

Definition 4.6 *Semantics of past tense operators can be defined as follows:*

$\Lambda, n \models \bigcirc^{-1}\varphi$ iff $n > 0$ and $\Lambda, n-1 \models \varphi$

$\Lambda, n \models \varphi \mathcal{S} \psi$ iff there is a $k \leq n$ such that $\Lambda, k \models \psi$ and $\Lambda, i \models \varphi$ for all $k < i \leq n$.

Example 4.11 Consider the trains and controller from Example 4.9. The trains can never enter the tunnel at the same moment, so $\mathsf{A}\Box(crash \rightarrow \bigcirc^{-1}(t_1 \vee t_2))$, i.e. if there is a crash, then a train must have already been in the tunnel in the previous moment. The formula is equivalent to $\mathsf{A}\Box\neg(\neg(t_1 \vee t_2) \wedge \bigcirc crash)$ when we consider both formulae from the perspective of the initial point of the computation: it cannot happen that no train is in the tunnel and in the next state the trains crash.[3] □

Example 4.12 Another useful past time formula is $\neg\bigcirc^{-1}true$, that specifies the starting point in computation. For instance, we may want to require that no train is in the tunnel at the beginning: $\neg\bigcirc^{-1}true \rightarrow \neg t_1 \wedge \neg t_2$, which is initially equivalent to $\neg t_1 \wedge \neg t_2$, but states the fact explicitly and holds for all points in all computations. Also, tautology $\mathsf{A}\Box\Diamond^{-1}\neg\bigcirc^{-1}true$ makes it clear that we deal with finite past in ATEL-R*. □

4.5.2 Knowledge vs. Observations

It can be interesting to reason about observations in ATEL-R*, too. We can embed ATOL in ATEL-R* in the following way:

Definition 4.7 *For all* $\Theta = CO, EO$ *or* DO*:*

$\Lambda, n \models \Theta_A\, \varphi$ iff for every Λ', n' such that $\Lambda'[n'] \sim_A^{\Theta} \Lambda[n]$ we have $\Lambda', n' \models \varphi$

$\Lambda, n \models \langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\varphi$ iff there is a uniform strategy with no recall V_A such that for every Λ', n', for which $\Lambda'[n'] \sim_\Gamma^{\Theta} \Lambda[n]$ and Λ' is consistent with V_A, we have $\Lambda', n' \models \varphi$.

Operators for memoryless strategies, identified by agents with recall ($\langle\langle A\rangle\rangle^{\bullet}_{\mathcal{K}(\Gamma)}$, $\mathcal{K} = C, E, D$*) and vice versa (*$\langle\langle A\rangle\rangle_{\Theta(\Gamma)}$, $\Theta = CO, EO, DO$*) can also be added in a straightforward way.*

[3] For a precise definition and more detailed discussion of *initial equivalence*, consult for instance (Laroussinie and Schnoebelen, 1995).

The explicit distinction between knowledge and observations can help to clarify a few things. The first one is more philosophical: an agent knows what he can see *plus* what he can remember to have seen. Or – more precisely – knowledge is what we can deduce from our present and past observations, provided we are given sufficient observational abilities (in the ontological sense, i.e. we can name what we see).

Proposition 4.7 *Suppose our language is rich enough to identify separate states, i.e. the set of propositions Π includes a proposition α_q for every state $q \in Q$, such that α_q is true only in q (since the set of states is always finite, we can always add such propositions to Π for each particular model). Then for every formula φ there exist formulae $\varphi'_1, \varphi''_1, \ldots, \varphi'_n, \varphi''_n$, such that $\varphi'_1, \ldots, \varphi'_n$ contain no epistemic operators with recall (K, C, E, D), and $\varphi'_i \wedge \varphi''_i \rightarrow \varphi$ for every i, and:*

$$\bigvee_{i=1..n} K_a\varphi \equiv (Obs_a\varphi'_i \wedge \bigcirc^{-1}K_a\bigcirc\varphi''_i) \vee (\neg\bigcirc^{-1}true \wedge Obs_a\varphi).$$

This implies that, in every situation, $K_a\varphi$ can be rewritten to some formula $Obs_a\varphi'_i \wedge \bigcirc^{-1}K_a\bigcirc\varphi''_i$ unless we are at the beginning of a run – then it should be rewritten to $Obs_a\varphi$.

Proof: Consider formulae $\varphi'_i \equiv \neg Obs_a\neg\alpha_{q_i}$ and $\varphi''_i \equiv \neg Obs_a\neg\alpha_{q_i} \rightarrow \varphi$, one pair for each state $q_i \in Q$. Let Λ, n be a computation and a position in it, and $\Lambda[n] = q_j$ current state of the computation. Suppose that $\Lambda, n \models K_a\varphi$; then for every Λ' such that $\Lambda'_{|n} \approx_a \Lambda_{|n}$, we have that $\Lambda', n \models \varphi$. Note that $\neg Obs_a\neg\alpha_{q_j}$ is true exactly in the states belief-accessible for a from q_j, so $\Lambda, n \models Obs_a(\neg Obs_a\neg\alpha_{q_j})$. Now, $\Lambda'_{|n-1} \approx_a \Lambda_{|n-1}$ and $\Lambda', n \models \neg Obs_a\neg\alpha_{q_j}$ imply that $\Lambda'_{|n} \approx_a \Lambda_{|n}$, so $\Lambda', n-1 \models \bigcirc(\neg Obs_a\neg\alpha_{q_j} \rightarrow \varphi)$ and hence $\Lambda, n \models \bigcirc^{-1}K_a\bigcirc(\neg Obs_a\neg\alpha_{q_j} \rightarrow \varphi)$. Finally, $\neg Obs_a\neg\alpha_{q_i}$ and $\neg Obs_a\neg\alpha_{q_i} \rightarrow \varphi$ obviously imply φ.

$\Lambda, n \models \bigcirc^{-1}K_a\bigcirc(\neg Obs_a\neg\alpha_{q_j} \rightarrow \varphi)$ and $\Lambda, n \models Obs_a(\neg Obs_a\neg\alpha_{q_j})$ imply $\Lambda, n \models K_a\varphi$ in an analogous way. □

Example 4.13 The above proposition can be illustrated with the systems in Figure 4.10. Consider path q_2q_6 for example. The agent must have known in q_2 that he was in q_1 or q_2 and therefore in the next step he can be in either q_5 or q_6. Now, in q_6 he can observe that the current state is q_6 or q_7, so it must be q_6 in which p holds. Note that the agent's ontology is too poor in system (A): he cannot express with the available language the differences he can actually see. Sufficient vocabulary is provided in Figure 4.10(B): for instance, when q_6 is the current state, $K_a\,p$ can be always rewritten as

$$Obs_a\neg Obs_a\neg q6 \wedge \bigcirc^{-1}K_a\bigcirc(\neg Obs_a\neg q6 \rightarrow p)$$

and of course $\neg Obs_a\neg q6 \wedge (\neg Obs_a\neg q6 \rightarrow p) \rightarrow p$. □

4.5.3 Complete Information vs. Uniform Strategies

Let $\langle\langle A\rangle\rangle\Phi$ denote that A have a complete information strategy to enforce Φ - like in ATL and original ATEL*. Relationship analogous to Proposition 4.7 can be shown between the incomplete and complete information cooperation modalities. This one is not past-, but future-oriented, however.

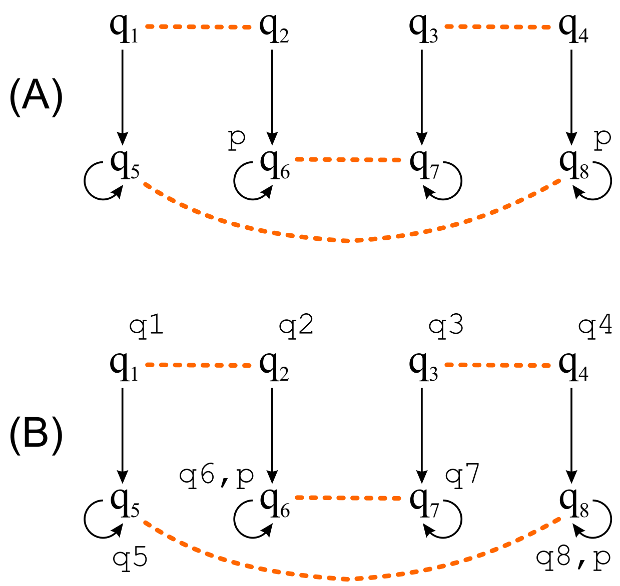


Figure 4.10: Knowledge vs. observations: with and without the vocabulary

Proposition 4.8 $\langle\langle A\rangle\rangle\Phi \equiv \langle\langle A\rangle\rangle_{C(\varnothing)}\bigcirc\langle\langle A\rangle\rangle\bigcirc^{-1}\Phi$. *In other words, having a complete information strategy is equivalent to having a uniform strategy that can be hinted at every step by an omniscient observer.*

A similar property can be shown for agents with no recall:

Proposition 4.9 $\langle\langle A\rangle\rangle^{\bullet}\Phi \equiv \langle\langle A\rangle\rangle^{\bullet}_{CO(\varnothing)}\bigcirc\langle\langle A\rangle\rangle^{\bullet}\bigcirc^{-1}\Phi$.

4.5.4 More Examples

Several further examples for ATEL-R* are presented below.

Example 4.14 For the variable client/server system from Examples 3.3 and 4.7, recall of the past adds nothing to the agents' powers:

- $\langle\langle s\rangle\rangle_{K(s)}\varphi \to \langle\langle s\rangle\rangle^{\bullet}_{Obs(s)}\varphi$,
- $\langle\langle c\rangle\rangle_{K(c)}\varphi \to \langle\langle c\rangle\rangle^{\bullet}_{Obs(c)}\varphi$,
- $K_s\varphi \to Obs_s\varphi$ etc.

This is because each state can be reached from all the other ones in a single step. Thus, knowledge of the previous positions in the game does not allow for any elimination of possible alternatives. Obviously, in a realistic setting, the agents would remember not only their local states from the past, but also the decisions they made – and that would improve the client's epistemic capacity. □

Example 4.15 Consider a model for the variable client and server, extended in a similar way as in Example 4.8 (in which every player remembers his last decision). For this system, the client starts to have complete knowledge of the situation as soon as x is assigned the value of 2:

- $\mathsf{A}\Box(x = 2 \rightarrow \mathsf{A}\Box(K_c(x = 0) \vee K_c(x = 1) \vee K_c(x = 2)))$;
- note that still $\mathsf{A}\Box(\neg Obs_c(x = 0) \wedge \neg Obs_c(x = 1))$.

On the other hand, the server gains only some knowledge of the past. If he has been rejecting the claims all the time, for instance, he knows that at the beginning the value of $\mathtt{x}$ must have been the same as now:

- $\Box^{-1}rej \rightarrow K_s(x = 0 \rightarrow \Box^{-1}(\neg\bigcirc^{-1}true \rightarrow x = 0))$ etc.

□

Example 4.16 Some properties of the train controller from Example 4.9 can be analyzed through formulae of ATEL-R*:

- $t_i \rightarrow K_c\bigcirc^{-1}\neg a_i$: every time a train is in the tunnel, c knows at least that in the previous moment it was not away;
- the controller is still unable to accomplish its mission: $\neg\langle\langle c\rangle\rangle_{K(c)}\Box\neg crash$, but...
- $a_1 \wedge a_2 \rightarrow \langle\langle c\rangle\rangle_{K(c)}(\Box\neg(a_1 \wedge a_2 \wedge \bigcirc(w_1 \wedge w_2)) \rightarrow \Box\neg crash)$. Suppose the trains never enter "the waiting zone" simultaneously and both are initially away – then c can finally keep them from crashing. The strategy is to immediately grant the green light to the first train that enters the zone, and keep it until the train is away again – then switch it to the other one if it has already entered the zone, and so on;
- also, if c is allowed to remember his last decision (i.e. the model is modified in the same way as in previous examples), then c knows who is in the tunnel: $\mathsf{A}\Box(K_c t_i \vee K_c\neg t_i)$ in the new model. In consequence, c can keep the other train waiting and avoid crash as well.

□

Example 4.17 Consider *IF* games again (see Example 4.10). An interesting variation on the theme can be to allow that a game is played repeatedly a (possibly infinite) number of times. For instance, we can have formula Υ_1 defined as a fixed point: $\Upsilon_1 \equiv \forall x\exists y/x\ (x \neq y \vee \Upsilon_1)$, which means that the game of $\forall x\exists y/x\ x \neq y$ should be played until Verifier wins. The structure of this game is presented in Figure 4.11.

- Verifier still cannot be guaranteed that he eventually wins: $S[\Upsilon_1], q_0 \ldots, 0 \models \neg\langle\langle V\rangle\rangle_{K(V)}\Diamond win$;
- this time around, however, V's success is much more likely: for each strategy of his, he fails on one path out of infinitely many possible (and Falsifier has to make up his mind *before* V). Intuitively, the probability of eventually bringing about win is 1, yet we do not see how this issue can be expressed in ATEL-R* or ATOL at present;

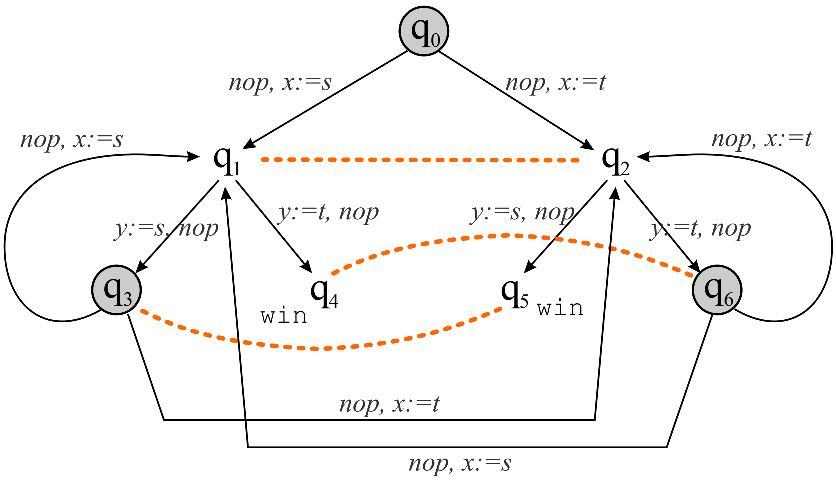


Figure 4.11: Game structure $S[\Upsilon_1]$ for game $\Upsilon_1 \equiv \forall x \exists y/x\ (x \neq y \vee \Upsilon_1)$

- note that in an analogous model for $\forall x \forall z \exists y/x\ x \neq y$ we have $S[\forall x \forall z \exists y/x\ x \neq y], q_0 \ldots, 0 \models \langle\langle V \rangle\rangle_{K(V)} \Diamond win$, yet this is only because the semantics of ATEL-R* does not treat $\sim_V$ as the epistemic accessibility relation, but rather as a basis from which the relation is generated. Hence, it allows V to remember the value of x anyway – which shows that $S[\forall x \forall z \exists y/x\ x \neq y]$ is not a suitable ATEL-R* model for the formula (although it is still an appropriate ATOL model);

- in order to encode the new game in ATEL-R*, we should split Verifier into two separate players V_1 and V_2. V_1 makes the move at the first and second steps and has a complete information about the state of the environment; V_2 does not see the Falsifier's choice for x at all. What we should ask about then is: $\langle\langle V_1 \rangle\rangle_{K(V_1)} \bigcirc \bigcirc \langle\langle V_2 \rangle\rangle_{K(V_2)} \Diamond win$, which naturally does not hold;

- the above shows that ATOL is much closer to the spirit of *IF* games than ATEL-R*. Why should we care about ATEL-R* for modeling *IF* games at all? Well, consider game $\forall x \Upsilon_2$, where $\Upsilon_2 \equiv \exists y/x\ (x \neq y \vee \Upsilon_2)$; the structure of the game is shown in Figure 4.12. In ATEL-R*, Verifier has a simple winning strategy: first try $y := s$, and the next time $y := t$, and he is bound to hit the appropriate value – hence, $S[\Upsilon_2], q_0 \ldots, 0 \models \langle\langle V \rangle\rangle_{K(V)} \Diamond win$. At the same time, V has no memoryless strategy: $S[\Upsilon_1], q_0 \ldots, 0 \models \neg \langle\langle V \rangle\rangle^{\bullet}_{Obs(V)} \Diamond win$, because he loses the knowledge what he did with y last time every time he uses y again. In a sense, $\langle\langle V \rangle\rangle_{K(V)}$ is closer to the way variables are treated in mathematical logic than $\langle\langle V \rangle\rangle^{\bullet}_{Obs(V)}$: in $\exists y \exists y\ \varphi$ both quantifiers refer to *different* variables that have the same name only incidentally.

□

Proposition 4.10 *Finally, the following formulae are examples of* ATEL-R* *tautologies:*

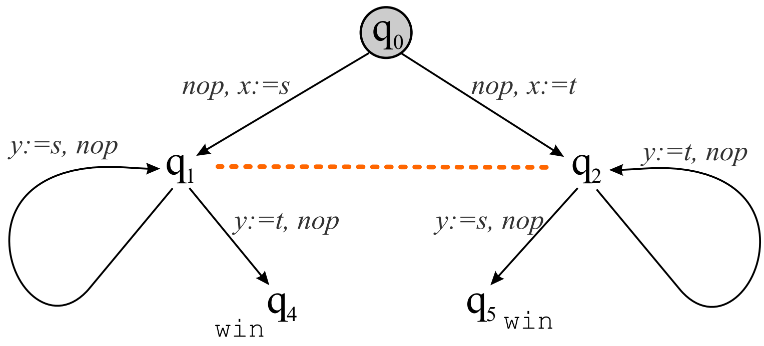


Figure 4.12: Game structure $S[\forall x \Upsilon_2]$

1. $\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\Phi \rightarrow \mathcal{K}_\Gamma\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\Phi$ *: if* Γ *are able to identify* A*'s strategy to bring about* Φ*, then they also know that* A *have such a strategy;*

2. $\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\Phi \rightarrow \mathcal{K}_\Gamma\langle\langle A\rangle\rangle_{CO(\varnothing)}\Phi$ *: if* A *have a strategy* de re*, then they have a strategy* de dicto*;*

3. *having a uniform strategy implies having a complete information strategy:* $\langle\langle A\rangle\rangle_{CO(\varnothing)}\Phi \rightarrow \langle\langle A\rangle\rangle\Phi$*;*

4. $\langle\langle A\rangle\rangle^{\bullet}_{\mathcal{K}(\Gamma)}\Phi \rightarrow \langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\Phi$ *: memoryless strategies are special cases of strategies with recall.*

4.5.5 Expressivity and Complexity of ATEL-R* and its Subsets

ATEL-R* logic, as defined here, subsumes ATL* and the original ATEL*, as well as Schobbens's ATL$_{ir}$*, ATL$_{iR}$* and ATL$_{Ir}$* logics from (Schobbens, 2003). One interesting issue about ATL*, ATL$_{ir}$*, ATL$_{iR}$* and ATL$_{Ir}$* is that they do not seem to be expressible by each other on the language level.[4] This is why we decided to include separate operators for each relevant perspective to epistemic and strategic abilities of agents.

The complexity results for "vanilla" ATEL-R are rather discouraging. Even parts of it are already intractable: ATL$_{ir}$ is NP-complete (Schobbens, 2003), and ATL$_{iR}$ (based on cooperation modalities for incomplete information and perfect recall) is generally believed to be undecidable, although no proof for it exists yet. We would like to stimulate a systematic investigation of the issue by extending the notation from (Emerson and Halpern, 1986). Let $B\ (P_1, P_2, \ldots \mid T_1, T_2, \ldots \mid M_1, M_2, \ldots)$ be the branching time logic with path quantifiers $P_1, P_2, \ldots$, temporal operators $T_1, T_2, \ldots$ and other modalities $M_1, M_2, \ldots$. Every temporal operator must have a path quantifier as its immediate predecessor (like in CTL). Then:

1. $B\ (E \mid \bigcirc, \Box, \mathcal{U} \mid -)$ is CTL;

[4] Except for ATL and ATL$_{Ir}$ – but without the star – which are equivalent

2. $B\ (\langle\langle A\rangle\rangle \mid \bigcirc, \Box, \mathcal{U} \mid -)$ is ATL;

3. $B\ (\langle\langle A\rangle\rangle \mid \bigcirc, \Box, \mathcal{U} \mid CO_A, EO_A, DO_A)$ is the original version of ATEL from (van der Hoek and Wooldridge, 2002);

4. $B\ (\langle\langle A\rangle\rangle_{CO(\varnothing)} \mid \bigcirc, \Box, \mathcal{U} \mid CO_A, EO_A, DO_A)$ is the ATEL version from (Jamroga, 2003d);

5. $B\ (\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)} \mid \bigcirc, \Box, \mathcal{U} \mid CO_A, EO_A, DO_A)$ is ATOL (Section 4.4);

6. $B\ (\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}, \langle\langle A\rangle\rangle^{\bullet}_{\mathcal{K}(\Gamma)}, \langle\langle A\rangle\rangle, \langle\langle A\rangle\rangle^{\bullet} \mid \bigcirc, \Box, \mathcal{U}, \bigcirc^{-1}, \Box^{-1}, \mathcal{S} \mid C_A, E_A, D_A, CO_A, EO_A, DO_A)$ is "vanilla" ATEL-R.

The model checking problem can be solved in polynomial time for (3). On the other hand, the same problem for (5) is NP-complete (Proposition 4.2). Note that allowing for perfect recall strategies (but with memoryless strategy identification) does not make things worse: model checking for $B\ (\langle\langle A\rangle\rangle_{\Theta(\Gamma)} \mid \bigcirc, \Box, \mathcal{U} \mid CO_A, EO_A, DO_A)$ is NP-complete in the same way (hint: use the model checking algorithm for (3) and guess the right set of states from which A can uniformly get to the current "good" states every time function pre is invoked). It turns out that the authors of the original ATEL proposed the largest tractable member of the family to date. Whether anything relevant can be added to it seems an important question.

4.6 Final Remarks

In this chapter, we have tried to point out that – when one wants to reason about knowledge of agents defined via alternating transitions systems or concurrent game structures (aka multi-player game models) – it is important to distinguish the computational structure from the behavioral structure of the system, and to decide in what way the first one unravels into the latter. We argue that the initial approach to Alternating-time Temporal Epistemic Logic (van der Hoek and Wooldridge, 2002) offered too weak a notion of a strategy. In order to say that agent a can enforce a property φ, it was required that there existed a sequence of a's actions at the end of which φ held – whether he had knowledge to recognize the sequence was not taken into account. Moreover, even the requirement that the agent's strategy must be uniform proved too weak: it would still enable plans in which the agent was allowed to "guess" the opening action. We suggest that it is not enough that the agent knows that some strategy will help him out; it is more appropriate to require that the agent can identify the winning strategy itself. In other words, the agent should be required to have a strategy *de re* rather than *de dicto*. Under such a constraint, the agent "knows how to play".

This is still not enough to give the meaning of a cooperation modality for coalitional planning under uncertainty. Even if a group of agents can collectively identify a winning strategy, they are prone to fail in case there are other competing strategies as well. Thus, we propose several different operators instead of one to distinguish subtle cases here.

The assumption that agents can use the complete history to make their subsequent decisions is also investigated in this chapter. Two paradigms are studied here in consequence. First, agents can be assumed to have no (or only limited) memory. In this case, they make their decisions only on the basis of what they can *observe* (albeit in the broadest possible sense of the word); a language for ATL + Observations, dubbed ATOL, is proposed for specification of such agents. The other paradigm is formalized via a richer system, called Alternating-time Temporal Epistemic Logic with Recall (ATEL-R*). We believe that both approaches can be equally interesting and useful.

The research, reported here, offers only a step towards clarifying epistemic issues in ATL, and it leaves many questions open. For instance, although only recently a complete axiomatization for ATL has been given (Goranko and van Drimmelen, 2003), this is still unexplored area for ATEL and ATOL. Also, more non-trivial examples of game-like scenarios should be looked for, in which a logic of knowledge and time may help to reveal interesting properties, and which are good cases for automated planning via model checking.

Chapter 5

Obligations vs. Abilities of Agents

SYNOPSIS. *The story unveils step by step. In a surprisingly logical way. First, a number of languages and semantic structures came up to enable reasoning and modeling agents and their environments of action. We traced the origins and basic concepts behind these logics, classified them, related them to each other – and did away with most of them before Chapter 2 was over. Then, in an ingenious move that made our story more dramatic and more sophisticated at the same time, we brought to the fore Alternating-time Temporal Epistemic Logic – only to show in the next chapter that* ATEL *is not completely what it had seemed. Of course, it allowed us to investigate the nature of the problem, and propose ways of improvement. The focus of the thesis has thus shifted from investigating proposals of other researchers to more presentation of our own ideas; however, even the original ideas of ours arose through an attempt to improve an existing logical system. It seems a good time now to propose a* new *combination of modal logics, so that new, young researchers can too detect incongruities, propose remedies, and do their PhDs in consequence.*

Through a wealth of uplifting episodes, and a few sudden twists of action, we have finally come to the point where we are not afraid of discussing the ruthless notion of obligation. And we even try to confront it with abilities. Only one thing troubles our conscience: the story has been lacking agents so far. Real, full-bodied agents, that is. But wait – the *ultimate Agent 007, James Bond comes to the scene in this chapter.*

5.1 Introduction

Alternating-time Temporal Logic has been playing a major role in the material presented so far. ATL and its models present itself as quite a general means of talking and thinking about autonomous agents, their actions, interactions, and strategic abilities. Chapter 3 shows how the language can be extended with the epistemic notions of individual and collective knowledge. Chapter 4, on the other hand, demonstrates that such extensions are not always as easy and straightforward as they seem at the first glance. Alternating-time Temporal Epistemic Logic was defined as consisting of two orthogonal layers: the strategic layer (inherited from ATL), and the epistemic layer (taken directly from epistemic logic). Unfortunately, it turns out that the two layers are in fact *not* independent in reality: the strategic abilities of an agent, acting under uncertainty, *heavily* depend on his actual knowledge. However, the core idea of extending ATL with other modalities, referring to other aspects of agents and their communities, seems generic and potent.

In this chapter, we propose a concept of "deontic ATL" (or DATL in short). As deontic logic focuses on obligatory behaviors of systems and agents, and Alternating-time Temporal Logic enables reasoning about abilities of agents and teams, we believe it interesting and potentially useful to combine these formal tools in order to confront system requirements (i.e., obligations) with possible ways of satisfying them by actors of the game (i.e., abilities). This work is not intended as a definite statement on how logics of obligation and strategic ability should be combined. Rather, we intend it to stimulate discussion about such kinds of reasoning, and the models that can underlie it.

We begin by presenting the main concepts from deontic logic. Then, in Section 5.3, a combination of ATL and deontic logic is defined and discussed. Three different approaches to modeling obligations in a temporal context are discussed: global requirements on states of the system (i.e., requirements that deem some states "correct" and some "incorrect"), local requirements on states ("correctness" may depend on the current state), and temporal obligations, which refer to paths rather than states. We investigate (in an informal way) the perspectives offered by each of these approaches, and present several interesting properties of agents and systems that can be expressed within their scope. Some preliminary formal results are given in Section 5.4. In particular, we present a reduction of DATL model checking to model checking of pure ATL formulae, yielding a DATL model checking algorithm that is linear in the size of the input model (and quadratic in the complexity of the input formula). Combining it with the planning algorithm from Section 2.8 enables efficient planning for deontic goals as well.

The chapter builds on (Jamroga et al., 2004), a paper co-written with Wiebe van der Hoek and Michael Wooldridge from the University of Liverpool.

5.2 Deontic Logic: The Logic of Obligations

Deontic logic is the modal logic of obligations. It was originally proposed by Mally in 1926 – but his logic turned out to introduce nothing really new in the formal sense. The contemporary deontic logic dates back to 1950s and the works of von Wright (von Wright, 1951). A survey on deontic logic can be found in (Meyer and Wieringa, 1993b), and especially (Meyer and Wieringa, 1993a). The basic concepts: *obligation*, *permission* and *prohibition*, are expressed with modal operators:

- $\mathcal{O}\varphi$: "it ought to be that φ" or "it is obligatory that φ",
- $\mathcal{P}\varphi$: "it is permitted that φ", and

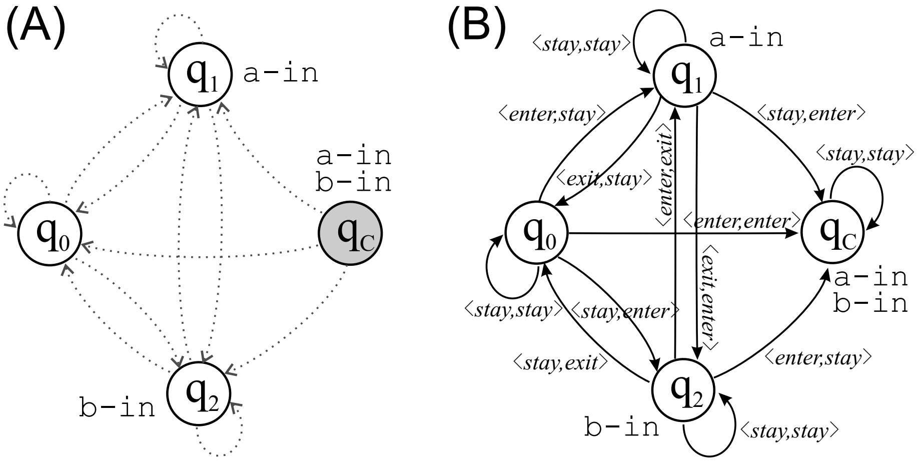


Figure 5.1: (A) Critical section example: the trains and the tunnel. Dotted lines display the deontic accessibility relation. (B) The trains revisited: temporal and strategic structure

- $\mathcal{F}\varphi$: "it is forbidden that φ".

It is usually accepted that the concept of obligation is primitive, and the other two are defined upon it:

- $\mathcal{F}\varphi \equiv \mathcal{O}\neg\varphi$,
- $\mathcal{P}\varphi \equiv \neg\mathcal{F}\varphi \equiv \neg\mathcal{O}\neg\varphi$,

although in some approaches obligations and permissions are treated independently (Alchourron, 1993; Fiadeiro and Maibaum, 1991).

5.2.1 Models and Semantics

In the traditional, von Wright's version of deontic logic, models are defined as Kripke structures with accessibility relation (or relations) $\mathcal{R}$ for modeling obligations (von Wright, 1951). A state q' such that $q\mathcal{R}q'$ is called a "perfect alternative" of state q; we can also say that q' is *acceptable* or *correct* from the perspective of q. As with the conventional semantics of modal operators, we define:

$$M, q \models \mathcal{O}\varphi \text{ iff for all } q' \text{ such that } q\mathcal{R}q' \text{ we have } M, q' \models \varphi.$$

Let us illustrate the idea with a simplified version of the "trains and tunnel" example (cf. Example 4.9).

Example 5.1 There are two trains: a and b; each can be inside a tunnel (propositions $\mathsf{a\text{-}in}$ and $\mathsf{b\text{-}in}$, respectively) or outside of it. The specification requires that the trains should not be allowed to be in the tunnel at the same time, because they will crash (so the tunnel can be seen as a kind of critical section): $\mathcal{F}(\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in})$ or, equivalently, $\mathcal{O}\neg(\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in})$. A model for the whole system is displayed in Figure 5.1A. □

The main idea behind this chapter is that, if we combine ATL and deontic logic, it may, among other things, allow us to express obligations about what coalitions should or should not achieve – without specifying *how* they achieve it (or refrain from it).

Example 5.2 Let us consider the tunnel example from a temporal (and strategic) perspective; a concurrent game structure for the trains and the tunnel is shown in Figure 5.1B. Using ATL, we have that $\langle\langle \mathbb{A}\mathrm{gt} \rangle\rangle \Diamond (\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in})$, so the system is physically able to display undesirable behavior. On the other hand, $\langle\langle a \rangle\rangle \Box \neg (\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in})$, i.e., train a can protect the system from violating the requirements. □

In this chapter, we propose to extend ATL with deontic operator $\mathcal{O}$ in order to investigate the interplay between agents' abilities and requirements they should meet. The resulting language, dubbed "Deontic ATL", or DATL in short, is defined in Section 5.3.

Substance of Obligations: Actions vs. States

Originally, obligations were given standard modal logic treatment, being modeled with accessibility relations that referred to *states* in the model – in the way we have just presented (von Wright, 1951, 1964; Anderson, 1958). Some recent approaches to deontic logic still use this perspective (Lomuscio and Sergot, 2003a,b). Meanwhile, actions have also been recognized as entities that can be obligatory, forbidden or permitted (Meyer, 1988; Alchourron, 1993), and this approach seems dominant in the current literature. It seems reasonable that the notions of moral (or legal) obligation, permission and prohibition should be in most cases related to actions one ought to (is allowed to, is forbidden to) execute, rather than to obligatory (acceptable, forbidden) states of the system. We believe, however, that the former stance still makes sense, especially when we treat deontic statements as referring to preservation (or violation) of some constraints one would like to impose on a system or some of its components (like integrity constraints in a database). In this sense, deontic modalities may refer to *requirements*: specification requirements, design requirements, security requirements etc. – an approach that has been already suggested in (Wieringa et al., 1989; Wieringa and Meyer, 1993; Broersen, 2003), albeit in different contexts. Thus, we will interpret $\mathcal{O}\varphi$ as "φ is required" rather than "φ ought to be" throughout the rest of the chapter. This approach allows to put all *physically* possible states of the system in the scope of the model, and to distinguish the states that are "correct" with respect to some criteria, thus enabling reasoning about possible faults and fault tolerance of the system (Wieringa and Meyer, 1993).

Locality and Individuality of Obligations

Let us go back to the trains and the tunnel from Example 5.1. Note that the set of perfect alternatives is the same for each state q in Figure 5.1A. In other words, the acceptability of situations is *global* and does not depend on the current state. Thus, the semantic representation can in fact be much simpler: it is sufficient to mark the states that *violate* the requirements with a special "violation" atom V (Anderson, 1958; Meyer, 1988). Or, equivalently, mark the forbidden states with "red", and the acceptable states with "green" (Lomuscio and Sergot, 2003a). Then the accessibility relation $\mathcal{R}$ can be defined as:

$$q\mathcal{R}q' \text{ iff } q' \not\models V.$$

Alternatively, we can use the following semantic rule:

$$M, q \models \mathcal{O}\varphi \text{ iff for all } q' \text{ such that } q' \not\models V \text{ we have } M, q' \models \varphi \text{ – or:}$$
$$M, q \models \mathcal{O}\varphi \text{ iff for all } q' \text{ we have } M, q' \models \neg V \rightarrow \varphi.$$

Using a more elaborate accessibility relation machinery makes it possible, in general, to model requirements that are *local* with respect to the current state. Is it necessary? In many application areas perhaps not. We argue in Section 5.3.3, however, that local obligations can provide a means for specifying requirements that evolve in time. Also, they can be used to specify exception handling in situations when full recovery of the system is impossible.

Another dimension of classifying obligations is their *individuality*. The accessibility relation can define the requirements for the whole system, or there can be many relations, specifying different requirements for each process or agent (Lomuscio and Sergot, 2003a). The requirements in Example 5.1, for instance, are universal rather than individual: they apply to the whole system. However, it may make sense to specify that the train b is required to avoid the tunnel at all (because, for instance, the tunnel is too narrow for it): $\mathcal{O}_b\neg\mathsf{b\text{-}in}$.

Paradoxes

Many paradoxes have been listed for various instances of deontic logic – cf. (Meyer and Wieringa, 1993a), for instance. We believe that (at least some of) the paradoxes are due to confusing various kinds of obligations, prohibitions etc. that are uttered with the same words in natural language, but their inherent meaning is in fact different. For example, one may confuse dynamic vs. static properties that ought to be satisfied (i.e. *actions* that ought to be executed vs. *states* the system should be in), ending up with *Ross's Paradox*, *Penitent's Paradox*, the paradox of *no contradictory obligations* etc. One may also confuse properties that ought to hold all the time vs. the ones that must hold at some future moment etc. as long as the temporal perspective is implicit (*Good Samaritan Paradox*). Defining permission as the dual of obligation (i.e. as a mere statement that φ *might* be morally acceptable, while e.g. reading "permission" as authorization suggests that φ is *proclaimed* to be acceptable) leads to much confusion too (*no free choice permission* paradox). There have been some attempts to clarify these issues. Alchourron (1993), for instance, makes an explicit distinction between positive permission (i.e. things that have been explicitly permitted) and negative permission (i.e. things that are merely not forbidden), and analyzes the formal relationship between these two concepts. Moreover, it is sometimes suggested that the use of deontic modalities should be restricted only to actions (action terms) and not to static properties.

We are not going do dig deep into these issues in this thesis. To avoid confusion, we will interpret deontic sentences as referring to *system requirements* (specification requirements, design requirements, security requirements etc.) that express which states of the system are considered *correct* in a given situation. Moreover, the only notion of permission, discussed in this chapter, is represented by the $\mathcal{UP}$ operator for "unconditionally permitted" situations (defined in Section 5.3.1). Thus, we leave the issue of permissions in general – and a discussion of the $\mathcal{P}$ operator – out of the scope of the thesis.

5.2.2 Combining Deontic Perspective with Other Modalities

The combination of deontic logic with temporal and dynamic logics has been investigated at length in the literature. A well-known reduction of deontic operators to dynamic logic was proposed in (Meyer, 1988):

- $\mathcal{F}\alpha \equiv [\alpha]V$,
- $\mathcal{P}\alpha \equiv \neg\mathcal{F}\alpha \equiv \langle\alpha\rangle\neg V$,
- $\mathcal{O}\alpha \equiv \mathcal{F}(-\alpha) \equiv [-\alpha]V$, where "$-\alpha$" stands for "not-doing α".

It turned out that embedding deontic concepts in dynamic logic not only enabled to express and investigate the interplay between obligations and time and actions, but it also cuts off some of the paradoxes. Another body of work proposes how deontic specifications can be reduced to temporal specifications (van Eck, 1982; Fiadeiro and Maibaum, 1991), while in (Maibaum, 1993) a reduction of deontic specifications to temporal ones via a kind of dynamic logic ("deontic action logic") is suggested. Finally, Dignum and Kuiper (1997) add temporal operators to dynamic deontic logic which serves as a basis. "Artificial social systems" and "social laws" for multiple agents acting in time (Moses and Tennenholz, 1990; Shoham and Tennenholz, 1992, 1997; Moses and Tennenholz, 1995) also contribute to the field in a broad sense.

In addition, combinations of deontic and epistemic logics have been investigated, too. Bieber and Cuppens (1993) proposed such a combination for the purpose of security analysis, and Moses and Tennenholz (1995) included epistemic operators and accessibility relations in their logical system for reasoning about artificial social systems. A generic concept of deontic interpreted systems was investigated in (Lomuscio and Sergot, 2002, 2003a,b). Related work concerns also extending the BDI framework (beliefs, desires and intentions) with obligations (Broersen et al., 2001a,b).

Finally, a recent proposal (van der Hoek et al., 2004) combines the deontic and strategic perspectives, applying the concept of social laws to ATL: behavioral constraints (specific model updates) are defined for ATL models, so that some objective can be satisfied in the updated model. Since that paper deals with similar territory as the ideas presented here, we discuss their relationship in more detail in Section 5.3.5.

5.3 Deontic ATL

In this section, we extend ATL with deontic operators. We follow the definition with an informal discussion on how the resulting logic (and its models) can help to investigate the interplay between agents' abilities and requirements that the system (or individual agents) should meet.

5.3.1 Syntax and Semantics

The combination of deontic logic and ATL proposed here is technically straightforward: the new language consists of both deontic and strategic formulae, and models include the temporal transition function and deontic accessibility relation as two independent layers. Thus, the recursive definition of DATL formulae is:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \mathcal{O}_A\varphi \mid \mathcal{UP}_A\varphi \mid \langle\langle A\rangle\rangle\bigcirc\varphi \mid \langle\langle A\rangle\rangle\Box\varphi \mid \langle\langle A\rangle\rangle\varphi_1\mathcal{U}\varphi_2$$

where $A \subseteq \mathbb{A}\mathrm{gt}$ is a set of agents.

Models for DATL can be called *deontic game structures*, and defined as tuples $M = \langle \mathbb{A}\mathrm{gt}, Q, \Pi, \pi, Act, d, \delta, \mathbb{R}\rangle$, where:

- $\mathbb{A}\mathrm{gt}$ is a (finite) set of all *agents*, and Q is a non-empty set of *states*,
- Π is a set of (atomic) *propositions*, and $\pi : Q \to \mathcal{P}(\Pi)$ is their *valuation*;
- Act is a set of actions, and $d : Q \times \mathbb{A}\mathrm{gt} \to \mathcal{P}(Act)$ is a function that returns the decisions available to player a at state q;
- a complete tuple of decisions $\langle\alpha_1, ..., \alpha_k\rangle \subseteq d_q(a_1) \times ... \times d_q(a_k)$ from all the agents in state q implies a deterministic transition according to the transition function $\delta(q, \alpha_1, ..., \alpha_k)$;
- finally, $\mathbb{R} : \mathcal{P}(\mathbb{A}\mathrm{gt}) \to \mathcal{P}(Q \times Q)$ is a mapping that returns a deontic accessibility relation $\mathcal{R}_A$ for every group of agents A.

The semantic rules for $p, \neg\varphi, \varphi \vee \psi, \langle\langle A\rangle\rangle\bigcirc\varphi, \langle\langle A\rangle\rangle\Box\varphi, \langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$ are inherited from the semantics of ATL (cf. Chapter 2), and the truth of $\mathcal{O}_A\varphi$ is defined in a similar way as in the version of dynamic logic presented in Section 5.2.1. We also propose a new deontic operator: $\mathcal{UP}\varphi$, meaning that "φ is unconditionally permitted", i.e., whenever φ holds, we are on the correct side of the picture. This new modality closely resembles the "knowing at most" notion from epistemic logic (Levesque, 1990).

$M, q \models p$	iff	$p \in \pi(q)$, for an atomic proposition p;
$M, q \models \neg\varphi$	iff	$M, q \not\models \varphi$;
$M, q \models \varphi \vee \psi$	iff	$M, q \models \varphi$ or $M, q \models \psi$;
$M, q \models \langle\langle A\rangle\rangle\bigcirc\varphi$	iff	there exists a collective strategy F_A such that for every computation $\Lambda \in out(q, F_A)$ we have $M, \Lambda[1] \models \varphi$;
$M, q \models \langle\langle A\rangle\rangle\Box\varphi$	iff	there exists a collective strategy F_A such that for every $\Lambda \in out(q, F_A)$ we have $M, \Lambda[i] \models \varphi$ for every $i \geq 0$;
$M, q \models \langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$	iff	there exists a collective strategy F_A such that for every $\Lambda \in out(q, F_A)$ there is $i \geq 0$ such that $M, \Lambda[i] \models \psi$ and for all j such that $0 \leq j < i$ we have $M, \Lambda[j] \models \varphi$;

$$M, q \models \mathcal{O}_A\varphi \quad \text{iff} \quad \text{for every } q' \text{ such that } q\mathcal{R}_A q' \text{ we have } M, q' \models \varphi;$$
$$M, q \models \mathcal{UP}_A\varphi \quad \text{iff} \quad \text{for every } q' \text{ such that } M, q' \models \varphi \text{ we have } q\mathcal{R}_A q'.$$

Operator $\mathcal{UP}_A$ – among other things – helps to characterize the *exact* set of "correct" states, especially in the case of local requirements, where the property of a state being "correct" depends on the current state of the system, and therefore cannot be characterized with an additional single proposition.

In principle, it should be possible that the requirements on a group of agents (or processes) are independent from the requirements for the individual members of the group (or its subgroups). Thus, we will not assume any specific relationship between relations $\mathcal{R}_A$ and $\mathcal{R}_{A'}$, even if $A' \subseteq A$. We propose only that a system can be identified with the complete group of its processes, and therefore the requirements on a system as a whole can be defined as: $\mathcal{O}\varphi \equiv \mathcal{O}_{\mathbb{A}\mathrm{gt}}\varphi$. In a similar way: $\mathcal{UP}\varphi \equiv \mathcal{UP}_{\mathbb{A}\mathrm{gt}}\varphi$.

5.3.2 Dealing with Global Requirements

Let us first consider the simplest case, i.e., when the distinction between "good" and "bad" states is global and does not depend on the current state. Deontic game structures can in this case be reduced to concurrent game structures with "violation" atom V that holds in the states that violate requirements. Then:

$$M, q \models \mathcal{O}\varphi \text{ iff for all } q' \text{ such that } q' \not\models V \text{ we have } M, q' \models \varphi.$$

As we have both requirements and abilities in one framework, we can look at the former and then ask about the latter. Consider the trains and tunnel example from Figure 5.1B, augmented with the requirements from Figure 5.1A. Let us also assume that these requirements apply to all the agents and their groups, i.e., $\mathcal{R}_A = \mathcal{R}_{A'}$ for all $A, A' \subseteq \mathbb{A}\mathrm{gt}$; we will continue to assume so throughout the rest of the chapter, unless explicitly stated. As already proposed, the trains are required not to be in the tunnel at the same moment, because it would result in a crash: $\mathcal{O}(\neg(\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in}))$. Thus, it is natural to ask whether some agent or team can prevent the trains from violating the requirement: $\langle\langle A\rangle\rangle\Box\neg(\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in})$? Indeed, it turns out that both trains have this ability: $\langle\langle a\rangle\rangle\Box\neg(\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in}) \wedge \langle\langle b\rangle\rangle\Box\neg(\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in})$. On the other hand, if the goal of a train implies that it passes through the tunnel, the train is unable to "safeguard" the system requirements any more: $\neg\langle\langle a\rangle\rangle\neg(\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in})\,\mathcal{U}(\mathsf{a\text{-}in} \wedge \neg\mathsf{b\text{-}in})$.

In many cases, it may be interesting to consider questions like: does an agent have a strategy to always/eventually fulfill the requirements? Or, more generally: does the agent have a strategy to achieve his goal in the way that does not violate the requirements (or so that he can recover from the violation of requirements eventually)? We try to list several relevant properties of systems and agents below:

1. the system is *stable* (with respect to model M and state q) if $M, q \models \langle\langle\varnothing\rangle\rangle\Box\neg V$, i.e., no agent (process) can make it crash;

2. the system is *semi-stable* (with respect to model M and state q) if it will inevitably recover from any future situation: $M, q \models \langle\langle\varnothing\rangle\rangle\Box\langle\langle\varnothing\rangle\rangle\Diamond\neg V$;

3. agents A form a (collective) *guardian* in model M at state q if they can protect the system from any violation of the requirements: $M, q \models \langle\langle A \rangle\rangle \Box \neg V$;

4. A can *repair the system* in model M at state q if $M, q \models \langle\langle A \rangle\rangle \Diamond \neg V$;

5. A is a (collective) *repairman* in model M at state q if A can always repair the system: $M, q \models \langle\langle \varnothing \rangle\rangle \Box \langle\langle A \rangle\rangle \Diamond \neg V$;

6. finally, another (perhaps the most interesting) property is agents' ability to eventually achieve their goal (φ) without violating the requirements. We say that agents A can *properly enforce* φ in M, q if $M, q \models \langle\langle A \rangle\rangle (\neg V) \, \mathcal{U} (\neg V \wedge \varphi)$.

We will illustrate the properties with the following example. The world is in danger, and only the Prime Minister (p) can save it through giving a speech at the United Nations session and revealing the dangerous plot that threatens the world's future. However, there is a killer (k) somewhere around who tries to murder him before he presents his speech. The Prime Minister can be hidden in a bunker (proposition pbunk), moving through the city (pcity), presenting the speech ($\mathsf{pspeaks} \equiv \mathsf{saved}$), or... well... dead after being murdered (pdead). Fortunately, the Minister is assisted by James Bond (b) who can search the killer out and destroy him (we are very sorry – we would prefer Bond to arrest the killer rather than do away with him, but Bond hardly works this way...). The deontic game structure for this problem is shown in Figure 5.2. The Prime Minister's actions have self-explanatory labels (*enter, exit, speak* and *nop* for "no operation" or "do nothing"). James Bond can defend the Minister (action *defend*), look for the killer (*search*) or stay idle (*nop*); the killer can either shoot at the Minister (*shoot*) or wait (*nop*). The Minister is completely safe in the bunker (he remains alive regardless of other agents' choices). He is more vulnerable in the city (can be killed unless Bond is defending him at the very moment), and highly vulnerable while speaking at the UN (the killer can shoot him to death even if Bond is defending him). James Bond can search out and destroy the killer in a while (at any moment). It is required that the world is saveable ($\mathcal{O}\langle\langle \mathbb{A}\mathrm{gt} \rangle\rangle \Diamond \mathsf{saved}$) and this is the only requirement ($\mathcal{UP}\langle\langle \mathbb{A}\mathrm{gt} \rangle\rangle \Diamond \mathsf{saved}$). Note also that the world can be saved if, and only if, the Prime Minister is alive (because $\langle\langle \mathbb{A}\mathrm{gt} \rangle\rangle \Diamond \mathsf{saved}$ is equivalent $\neg \mathsf{pdead}$), and the two states that violate this requirement are marked accordingly (V, which is of course equivalent to pdead).

The system is neither stable nor semi-stable (the Minister can go to the UN building and get killed, after which the system has no way of recovering). Likewise, no agent can repair the system in states q_7, q_8, and hence there is no repairman. The Prime Minister is a guardian as long as he stays in the bunker: $\mathsf{pbunk} \rightarrow \langle\langle p \rangle\rangle \Box \neg \mathsf{pdead}$, because he can stay in the bunker forever. However, if he does so, he cannot save the world: $\neg \langle\langle p \rangle\rangle (\neg \mathsf{pdead}) \, \mathcal{U} (\neg \mathsf{pdead} \wedge \mathsf{saved})$. On the other hand, he can cooperate with Bond to properly save the world as long as he is initially out of the UN building: $(\mathsf{pbunk} \vee \mathsf{pcity}) \rightarrow \langle\langle p, b \rangle\rangle (\neg \mathsf{pdead}) \, \mathcal{U} (\neg \mathsf{pdead} \wedge \mathsf{saved})$ – he can get to the bunker, defended by Bond, and then wait there until Bond finds the killer; then he can go out to present his speech. Incidentally, there is one more guardian in the system – namely, the killer: $(\neg \mathsf{pdead}) \rightarrow \langle\langle k \rangle\rangle \Box \neg \mathsf{pdead}$, and also $(\neg \mathsf{pdead}) \rightarrow \langle\langle p, k \rangle\rangle (\neg \mathsf{pdead}) \, \mathcal{U} (\neg \mathsf{pdead} \wedge \mathsf{saved})$, so the Minister can alternatively pay the killer instead of employing Bond.

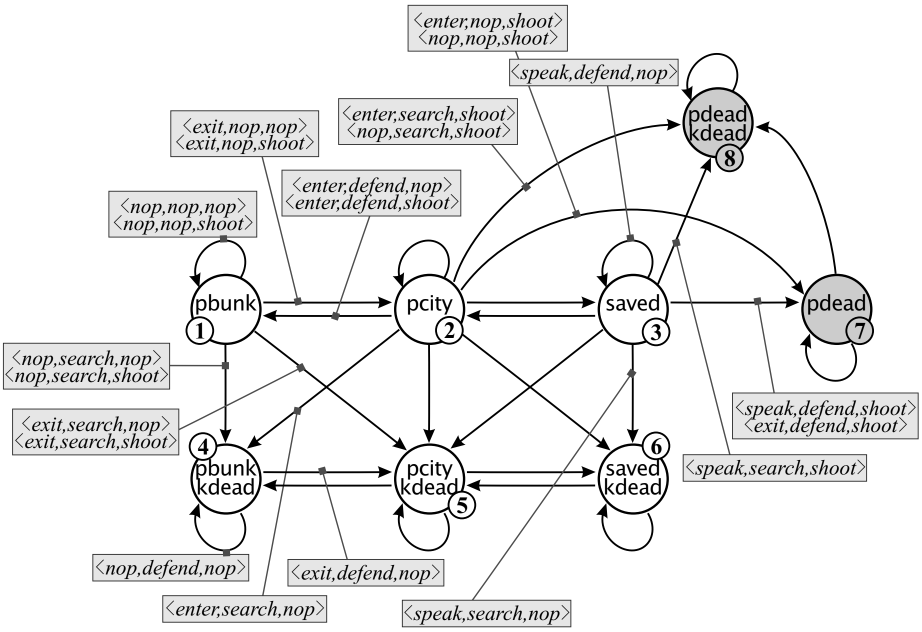


Figure 5.2: James Bond saves the world. The arrows show possible transitions of the system; some of the labels are omitted to improve readability. The states that violate the requirements are marked grey.

Remark 5.1 *Note that formula $\mathcal{O}\varphi \wedge \mathcal{UP}\varphi$ characterizes the exact set of correct states in the sense that $M, q \models \mathcal{O}\varphi \wedge \mathcal{UP}\varphi$ iff $\varphi \equiv \neg V$. Thus, $\mathcal{O}\varphi \wedge \mathcal{UP}\varphi$ can be seen as the deontic counterpart of the "only knowing" alias "all I know" operator from epistemic logic (Levesque, 1990).*

5.3.3 Local Requirements with Deontic ATL

A more sophisticated deontic accessibility relation may be convenient for modeling dynamics of obligations, for instance when the actors of the game can negotiate the requirements (e.g., deadlines for a conference submission). Alternatively, "localized" requirements can give a way of specifying *exception handling* in situations when a full recovery is impossible.

Example 5.3 Consider the modified "James Bond" example from Figure 5.3. The Prime Minister is alive initially, and it is required that he should be protected from being shot: $q_3 \models \neg \mathsf{pdead}$ and $q_3 \models \mathcal{O}\neg \mathsf{pdead}$. On the other hand, nobody except the killer can prevent the murder: $q_3 \models \langle\langle k \rangle\rangle \Box \neg \mathsf{pdead} \wedge \neg \langle\langle p, b \rangle\rangle \Box \neg \mathsf{pdead}$; moreover, when the president is dead, there is no way for him to become alive again ($\mathsf{pdead} \rightarrow \langle\langle \varnothing \rangle\rangle \Box \mathsf{pdead}$). Now, when the Minister is shot, a new requirement is implemented, namely it is required that either the Minister is resurrected or the killer is eliminated: $q_7 \models \mathcal{O}(\neg \mathsf{pdead} \vee \mathsf{kdead})$. Fortunately, Bond can bring about the latter:

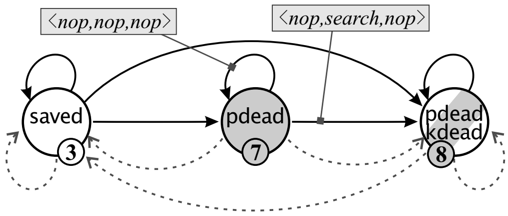


Figure 5.3: "James Bond saves the world" revisited: local requirements. Dotted lines define the deontic accessibility relation. Solid lines show possible transitions of the system.

$q_7 \models \langle\langle b\rangle\rangle\Diamond$kdead. Note that q_8 is unacceptable when the Minister is alive (q_3), but it becomes the only option when he has already been shot (q_7). □

Remark 5.2 *In a way, we are making the deontic accessibility relation serial in a very special sense, i.e., every state has at least one* reachable *perfect alternative now. We suggest to call this semantic property* effective seriality. *It is a well known fact that seriality of a modal accessibility relation corresponds to the D axiom, stating (in the case of obligations) that* $\neg\mathcal{O}_A\bot$ *or, equivalently,* $\mathcal{O}_A\varphi \to \neg\mathcal{O}_A\neg\varphi$ *(van der Hoek and Verbrugge, 2002). We conjecture that the effective seriality may correspond to the following axiom scheme:*

$$D_{EFF}\text{: } \mathcal{O}_A\varphi \to (\neg\mathcal{O}_A\neg\varphi \land \langle\langle \mathbb{A}\mathrm{gt}\rangle\rangle\Diamond\varphi)$$

or, equivalently:

$$D_{EFF}\text{: } (\mathcal{O}_A\varphi \land \mathcal{UP}_A\varphi) \to \langle\langle \mathbb{A}\mathrm{gt}\rangle\rangle\Diamond\varphi.$$

Similar properties of agents and systems to the ones from the previous section can be specified:

1. the system is *stable* in M, q if, given $M, q \models \mathcal{O}\varphi \land \mathcal{UP}\varphi$, we have $M, q \models \langle\langle\varnothing\rangle\rangle\Box\varphi$;

2. the system is *semi-stable* in M, q if, given that $M, q \models \mathcal{O}\varphi \land \mathcal{UP}\varphi$, we have $M, q \models \langle\langle\varnothing\rangle\rangle\Box(\varphi \to \langle\langle\varnothing\rangle\rangle\Diamond\varphi)$;

3. A form a *guardian* in M, q if, given $M, q \models \mathcal{O}\varphi \land \mathcal{UP}\varphi$, we have $M, q \models \langle\langle A\rangle\rangle\Box\varphi$;

4. A can *repair* the system in M, q if, given that $M, q \models \mathcal{O}\varphi \land \mathcal{UP}\varphi$, we have $M, q \models \langle\langle A\rangle\rangle\Diamond\varphi$;

5. group A is a *repairman* in M, q if, given that $M, q \models \mathcal{O}\varphi \land \mathcal{UP}\varphi$, we have $M, q \models \langle\langle\varnothing\rangle\rangle\Box\langle\langle A\rangle\rangle\Diamond\varphi$;

6a. A can *properly enforce* ψ in M, q if, given that $M, q \models \mathcal{O}_A\varphi \land \mathcal{UP}_A\varphi$, we have $M, q \models \langle\langle A\rangle\rangle\varphi\mathcal{U}(\varphi \land \psi)$. Note that this requirement is individualized now;

6b. A can *properly (incrementally) enforce* ψ in M, q if, given that $M, q \models \mathcal{O}_A\varphi \wedge \mathcal{UP}_A\varphi$, we have $M, q \models \varphi \wedge \psi$, or $M, q \models \varphi$ and A have a collective strategy F_A such that for every $\lambda \in out(q, F_A)$ they can properly (incrementally) enforce ψ in $M, \lambda[1]$.

The definitions show that many interesting properties, combining deontic and strategic aspects of systems, can be defined using semantic notions. At present, however, we do not see how they can be specified entirely in the object language.

5.3.4 Temporal Requirements

Many requirements have a temporal flavor, and the full language of ATL* allows to express properties of temporal paths as well. Hence, it makes sense to look at DATL*, where one may specify deontic temporal properties in terms of correct computations (rather than single states). In its simplest version, we obtain DTATL by only allowing requirements over temporal (path) subformulae that can occur within formulae of ATL:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle\langle A\rangle\rangle \bigcirc \varphi \mid \langle\langle A\rangle\rangle \Box\varphi \mid \langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2 \mid \mathcal{O}_A \bigcirc \varphi \mid \mathcal{O}_A \Box\varphi \mid \mathcal{O}_A \varphi_1 \mathcal{U} \varphi_2 \mid \mathcal{UP}_A \bigcirc \varphi \mid \mathcal{UP}_A \Box\varphi \mid \mathcal{UP}_A \varphi_1 \mathcal{U} \varphi_2.$$

Below we list several properties that can be expressed in this framework:

1. $\mathcal{O}\Diamond\langle\langle A\rangle\rangle\Box\varphi$: it is required that sometime in the future, coalition A gets the opportunity to guarantee φ forever,

2. $\mathcal{O}\Diamond(\langle\langle A\rangle\rangle\Diamond\varphi \wedge \langle\langle A\rangle\rangle\Diamond\neg\varphi)$: it is a requirement that eventually coalition A can determine φ;

3. the latter can be strengthened to $\mathcal{O}\Box(\langle\langle A\rangle\rangle\Diamond\varphi \wedge \langle\langle A\rangle\rangle\Diamond\neg\varphi)$, saying that it is an obligation of the system that there must always be opportunities for A to toggle φ as it wants.

Note that the definition of DTATL straightforwardly allows to express stability properties like

$$\mathcal{O}T\psi \rightarrow \langle\langle A\rangle\rangle T\psi$$

saying that A can bring about the temporal requirement $T\psi$.

Semantically, rather than being a relation between states, relation $\mathcal{R}_A$ is now one between states and computations (sequences of states). Thus, for any computation λ, $q\mathcal{R}_A\lambda$ means that λ is an ideal computation, given q. The semantics of temporal obligations and unconditional permissions can be defined as:

$M, q \models \mathcal{O}_A \bigcirc \varphi$	iff	for every λ such that $q\mathcal{R}_A\lambda$, we have $M, \lambda[1] \models \varphi$;
$M, q \models \mathcal{O}_A \Box \varphi$	iff	for every λ such that $q\mathcal{R}_A\lambda$, we have $M, \lambda[i] \models \varphi$ for all $i \geq 0$;
$M, q \models \mathcal{O}_A \varphi \mathcal{U} \psi$	iff	for every λ such that $q\mathcal{R}_A\lambda$, there is $i \geq 0$ such that $M, \lambda[i] \models \psi$ and for all $0 \leq j < i$ we have $M, \lambda[j] \models \varphi$.
$M, q \models \mathcal{UP}_A \bigcirc \varphi$	iff	for every λ such that $M, \lambda[1] \models \varphi$, we have $q\mathcal{R}_A\lambda$;
$M, q \models \mathcal{UP}_A \Box \varphi$	iff	for every λ such that $M, \lambda[i] \models \varphi$ for all $i \geq 0$, we have $q\mathcal{R}_A\lambda$;
$M, q \models \mathcal{UP}_A \varphi \mathcal{U} \psi$	iff	for every λ, such that $M, \lambda[i] \models \psi$ for some $i \geq 0$ and $M, \lambda[j] \models \varphi$ for all $0 \leq j < i$, we have $q\mathcal{R}_A\lambda$.

One of the most appealing temporal constraints is that of a deadline: team A should achieve property φ within a number (say n) of steps. This could be just expressed by $\mathcal{O}_A \bigcirc^n \varphi$: only these courses of action are acceptable, in which the deadline is met.[1] Note that the DATL obligation $\mathcal{O}(\langle\langle A\rangle\rangle \bigcirc)^n \varphi$ expresses a different property: here, A must be *able* to meet the deadline.

Fairness-like properties are also a very natural area to reason about deontic constraints. Suppose we have a resource p that can only be used by one agent at the time (and as long as a is using it, p_a is true). The constraint that every agent should eventually be able to use the resource is expressed by $\bigwedge_{a \in \Sigma} \mathcal{O}\Box\langle\langle a\rangle\rangle\Diamond p_a$ – or, if this is an obligation of a particular scheduler s, we could write $\mathcal{O}_s$ rather than $\mathcal{O}$. Finally, let us recall the ATL operator $[\![A]\!]\Phi \equiv \neg\langle\langle A\rangle\rangle\neg\Phi$ (coalition A cannot prevent φ from being the case). Formula $\mathcal{O}\Box(\langle\langle A\rangle\rangle\Diamond\varphi \rightarrow [\![A]\!]\Box(\varphi \rightarrow \langle\langle A'\rangle\rangle\Diamond\neg\varphi))$ says that only these courses of action are acceptable in which, might coalition A ever have a way to enforce φ, then it must "pass the token" to A' and give the other agents the ability to reverse this again.

Note also that DTATL formulae $\mathcal{UP}\psi$ express a kind of "the end justifies means" properties. For instance, $\mathcal{UP}\Diamond\mathsf{kdead}$ means that *every* course of action, which yields the killer dead, is acceptable.

5.3.5 Deontic ATL and Social Laws

We mentioned the two main streams in deontic logic, having either states of affairs or actions as their object of constraints. In Deontic ATL, one can express deontic requirements about *who is responsible* to achieve something, without specifying how it should be achieved. The requirement $\mathcal{O}\neg\langle\langle\{a, b\}\rangle\rangle\Diamond\mathsf{safe\text{-}open}$, for example, states that it should be impossible for a and b to bring about the disclosure of a safe in a bank. However, with c being a third employee, we might have $\mathcal{O}(\neg\langle\langle\{a, b\}\rangle\rangle\Diamond\mathsf{safe\text{-}open} \wedge \langle\langle\{a, b, c\}\rangle\rangle\Box\mathsf{safe\text{-}open})$: as a team of three, they *must* be able to do so. We can also express delegation, as in $\mathcal{O}_a\langle\langle b\rangle\rangle\Box\varphi$: authority a has the obligation that b can always bring about φ.

[1] $\mathcal{O}\bigcirc^n\varphi$ is not a DTATL formula, but the logic can be easily extended to include it.

A recent paper (van der Hoek et al., 2004) also addresses the issue of prescribed behavior in the context of ATL: behavioral constraints (specific model updates) are defined for ATL models, so that some objective can be satisfied in the updated model. The emphasis in (van der Hoek et al., 2004) is on how the effectiveness, feasibility and synthesis problems in the area of social laws (Moses and Tennenholz, 1990; Shoham and Tennenholz, 1992) can be posed as ATL model checking problems. One of the main questions addressed is: given a concurrent game structure M and a social law with objective φ (which we can loosely translate as $\mathcal{O}\varphi$), can we modify the original structure M into M', such that M' satisfies $\langle\!\langle \emptyset \rangle\!\rangle \Box \varphi$? In other words, we ask whether the overall system can be altered in such a way that it cannot but satisfy the requirements. The question whether certain coalitions are *able* to "act according to the law" is not addressed in (van der Hoek et al., 2004); the law is *imposed* on the system as a whole. On the other hand, we are interested in bringing requirements into the scope of ATL, so that one can ask questions about which courses of action are "correct", and what particular agents or coalitions can do about it. Can they enforce that no forbidden states will be achieved, for instance? This is a different question from whether some higher-order entity (e.g. the designer, system administrator etc.) can redefine the game so that the requirements always hold. Thus, the approach of that paper is prescriptive, while our approach here is rather descriptive.

The difference is also reflected in the semantics: here, requirements can be referred to via deontic sentences in the object level, and via modal accessibility relation on the semantic side. In (van der Hoek et al., 2004), the requirements (objectives) are expressed purely syntactically, and the focus is on model updates that can lead to a model in which every state satisfies them. Moreover, (van der Hoek et al., 2004) lacks explicit deontic notions in the object level.

An example of a requirement that cannot be imposed on the system as a whole, taken from (van der Hoek et al., 2004), is $p \wedge \langle\!\langle A \rangle\!\rangle \bigcirc \neg p$: property p is obligatory, but at the same time, A should be *able* to achieve $\neg p$. This kind of constraints could be used to model exceptional situations, such as: "it is obligatory that the emergency exit is not used, although at the same time people in the building should always be able to use it". Imposing such an overall objective upon a system means that our behavioral constraints should both rule out any possibility of $\neg p$ from the system, and retain the possibility of deviating from p in it – which is obviously impossible. It seems that our Deontic ATL covers a more local notion of obligation, in which $\mathcal{O}(p \wedge \langle\!\langle A \rangle\!\rangle \bigcirc \neg p)$ can well be covered in a non-trivial way.

Note that our "stability" requirements are similarly strong: in fact, the property of M being stable in state q, given that $M, q \models \mathcal{O}\varphi \wedge \mathcal{UP}\varphi$ (cf. Section 5.3.3), is equivalent to the *effectiveness* of (φ, β_I) in M, q (where β_I is the "identity" constraint, i.e. $\beta_I(\alpha) = Q$ for each action α). On the other hand, our "guardian" requirements are rather weak: to demand that every obligation $\mathcal{O}\varphi$ is implementable by a coalition does not yet guarantee that the system *does* behave well. In each particular case, we might be looking for something in between the universal guarantee and a coalitional efficiency with respect to constraint φ. And it is one of the features of Deontic ATL – that one can express many various stability requirements, making explicit who is responsible for what.

5.4 Axioms, Model Checking and Similar Stories

Let DL be the language of deontic logic. Then – if we do not have any mixing axioms relating the coalitional and the deontic operators – we obtain logic DATL as an *independent combination* of two modal logics: ATL $\oplus$ DL (Franceschet et al., 2000). Franceschet et al. give an algorithm for model checking such combinations, given two model checkers for each separate logic. However, two technical remarks are in order here. First, the formal results from (Franceschet et al., 2000) refer to combining *temporal* logics, while neither ATL nor DL is a temporal logic in the strictest sense. Moreover, the algorithm they propose for model checking of an independent combination of logics assumes that the models are finite (while there is no such assumption in our case). Nevertheless, polynomial model checking of DATL is possible, and we show how it can be done in Section 5.4.2, through a reduction of the problem to ATL model checking.[2]

5.4.1 Imposing Requirements through Axioms

Following the main stream in deontic logic, we can take every deontic modality to be KD – the only deontic property (apart from the K-axiom and necessitation for $\mathcal{O}_A$) being the D-axiom $\neg\mathcal{O}_A\bot$. An axiomatization of ATL has been recently shown in (Goranko and van Drimmelen, 2003). If we do not need any mixing axioms, then the axiomatization of DATL can simply consist of the axioms for ATL, plus those of DL.

Concerning the global requirements, note that endowing DATL models with the violation atom V is semantically very easy. Evaluating whether $\mathcal{O}\varphi$ is true at state q suggests incorporating a *universal modality* (Goranko and Passy, 1992) although some remarks are in place here. First of all, it seems more appropriate to use this definition of global requirements in *generated models* only, i.e., those models that are generated from some initial state q_0, by the transitions that the grand coalition $\mathbb{A}\mathrm{gt}$ can make. Otherwise, many natural situations may be unnecessarily hard to capture because of considering violations (or their absence) in unreachable states. As an example, suppose we have a system that has two *separate* subsystems: in the first subsystem (with q_1 as the initial state), we must drive in the continental style, while in the latter (with q_2 as the initial state) British traffic rules apply. Thus, starting from q_1, we violate the requirements while driving on the left hand side of the road ($V \equiv \mathsf{left}$), but when the system starts from q_2, driving on the right hand side is a violation of the law ($V \equiv \mathsf{prop}$). To specify one global requirement, we need additional propositions to identify each subsystem: $\mathcal{O}((\mathsf{british} \rightarrow \mathsf{left}) \wedge (\mathsf{continental} \rightarrow \mathsf{right}))$. Alternatively, we can opt for a more general solution, and define obligations in a system M with root q_0 as:

$$M, q \models \mathcal{O}\varphi \text{ iff } M, q_0 \models \langle\langle\varnothing\rangle\rangle\Box(\neg V \rightarrow \varphi).$$

Second, we note in passing that by using the global requirement definition of obligation, the $\mathcal{O}$ modality obtained in this way is a KD45 modality, which means that we

[2]Similar remark applies of course to ATEL in Chapter 3, which is an independent combination of ATL and epistemic logic.

inherit the properties $\mathcal{O}\varphi \rightarrow \mathcal{O}\mathcal{O}\varphi$ and $\neg\mathcal{O}\varphi \rightarrow \mathcal{O}\neg\mathcal{O}\varphi$, as was also observed in (Lomuscio and Sergot, 2003a). But also, we get mixing axioms in this case: every deontic subformula can be brought to the outmost level, as illustrated by the valid scheme

$$\langle\langle A\rangle\rangle\Diamond\mathcal{O}\varphi \leftrightarrow \mathcal{O}\varphi$$

(recall that we have $M, q \models \mathcal{O}\varphi$ iff $M, q' \models \mathcal{O}\varphi$, for all states q, q' and root q_0).

Some of the properties we have mentioned earlier in this chapter can constitute interesting mixing axioms as well. For instance, a minimal property for requirements might be

$$\mathcal{O}_A\varphi \rightarrow \langle\langle A\rangle\rangle\Diamond\varphi$$

saying that every coalition can achieve its obligations. Semantically, we can pinpoint such a property as follows. Let us assume that this is an axiom scheme, and the model is distinguishing (i.e., every state in the model can be characterized by some DATL formula). Then the scheme corresponds to the semantic constraint:

$$\forall q \exists F_A \forall \lambda \in out(q, F_A) : states(\lambda) \cap img(q, \mathcal{R}_A) \neq \varnothing$$

where $states(\lambda)$ is the set of all states from λ, and $img(q, R) = \{q' \mid qRq'\}$ is the image of q with respect to relation R. In other words, A can enforce that every possible computation goes through at least one perfect alternative of q.

Another viable mixing axiom is the $\mathrm{D}_{\mathrm{EFF}}$ axiom from Remark 5.2, that corresponds to "effective seriality" of the deontic accessibility relation.

5.4.2 Model Checking Requirements and Abilities

In this section, we present a satisfiability preserving interpretation of DATL into ATL. The interpretation is very close to the one from Section 3.4, which in turn was inspired by (Schild, 2000). The main idea is to leave the original temporal structure intact, while extending it with additional transitions to "simulate" deontic accessibility links. The simulation is achieved through new "deontic" agents: they can be passive and let the "real" agents decide upon the next transition (action $pass$), or enforce a "deontic" transition. More precisely, the "positive deontic agents" can point out a state that was deontically accessible in the original model (or, rather, a special "deontic" copy of the original state), while the "negative deontic agents" can enforce a transition to a state that was *not* accessible. The first ones are necessary to translate formulae of shape $\mathcal{O}_A\varphi$; the latter are used for the "unconditionally permitted" operator $\mathcal{UP}_A$.

As an example, let M be the deontic game structure from Figure 5.3, and let us consider formulae $\mathcal{O}_{\mathbb{A}\mathrm{gt}}\neg\mathsf{saved}$, and $\mathcal{UP}_{\mathbb{A}\mathrm{gt}}\mathsf{saved}$ and $\langle\langle k, b\rangle\rangle\bigcirc\mathsf{pdead}$ (note that all three formulae are true in M, q_3). We construct a new concurrent game structure M^{ATL} by adding two deontic agents: $r_{\mathbb{A}\mathrm{gt}}, \bar{r}_{\mathbb{A}\mathrm{gt}}$, plus "deontic" copies of the existing states: $q_3^{\mathsf{r}_{\mathbb{A}\mathrm{gt}}}, q_7^{\mathsf{r}_{\mathbb{A}\mathrm{gt}}}, q_8^{\mathsf{r}_{\mathbb{A}\mathrm{gt}}}$ and $q_3^{\bar{\mathsf{r}}_{\mathbb{A}\mathrm{gt}}}, q_7^{\bar{\mathsf{r}}_{\mathbb{A}\mathrm{gt}}}, q_8^{\bar{\mathsf{r}}_{\mathbb{A}\mathrm{gt}}}$ (cf. Figure 5.4). Agent $r_{\mathbb{A}\mathrm{gt}}$ is devised to point out all the perfect alternatives of the actual state. As state q_3 has only one perfect alternative (i.e., q_3 itself), $r_{\mathbb{A}\mathrm{gt}}$ can enforce the next state to be $q_3^{\mathsf{r}_{\mathbb{A}\mathrm{gt}}}$, provided that all other relevant agents remain passive.[3] In consequence, $\mathcal{O}_{\mathbb{A}gt}\mathsf{saved}$ translates as: $\neg\langle\langle r_{\mathbb{A}gt}, \bar{r}_{\mathbb{A}gt}\rangle\rangle\bigcirc(\mathsf{r}_{\mathbb{A}gt} \wedge \mathsf{saved})$. In other words, it is not possible that $r_{\mathbb{A}gt}$ points out an alternative of q_3 (while $\bar{r}_{\mathbb{A}gt}$ obediently passes), in which saved does *not* hold.

Agent $\bar{r}_{\mathbb{A}gt}$ can point out all the *imperfect* alternatives of the current state (for q_3, these are: $q_7^{\bar{\mathsf{r}}_{\mathbb{A}gt}}, q_8^{\bar{\mathsf{r}}_{\mathbb{A}gt}}$). Now, $\mathcal{UP}_{\mathbb{A}gt}\mathsf{saved}$ translates as $\neg\langle\langle r_{\mathbb{A}gt}, \bar{r}_{\mathbb{A}gt}\rangle\rangle\bigcirc(\bar{\mathsf{r}}_{\mathsf{A}} \wedge \mathsf{saved})$: $\bar{r}_{\mathbb{A}gt}$ cannot point out an unacceptable state in which saved holds, hence the property of saved guarantees acceptability. Finally, formula $\langle\langle k, b\rangle\rangle\bigcirc\mathsf{pdead}$ can be translated as $\langle\langle k, b, r_{\mathbb{A}gt}, \bar{r}_{\mathbb{A}gt}\rangle\rangle\bigcirc(\mathsf{act} \wedge \mathsf{pdead})$: the strategic structure of the model has remained intact, but we must make sure that both deontic agents are passive, so that a non-deontic transition (an "action" transition) is executed.

We present the whole translation below in a more formal way, and refer to Section 3.4 for a detailed presentation of the method and proofs of correctness.

Given a deontic game structure $M = \langle \mathbb{A}gt, Q, \Pi, \pi, Act, d, \delta, \mathbb{R}\rangle$ for a set of agents $\mathbb{A}gt = \{a_1, ..., a_k\}$, we construct the corresponding concurrent game structure $M^{\mathrm{ATL}} = \langle \mathbb{A}gt', Q', \Pi', \pi', Act', d', \delta'\rangle$ in the following manner:

- $\mathbb{A}gt' = \mathbb{A}gt \cup \mathbb{A}gt^r \cup \mathbb{A}gt^{\bar{r}}$, where $\mathbb{A}gt^r = \{r_A \mid A \subseteq \mathbb{A}gt, A \neq \varnothing\}$ is the set of "positive", and $\mathbb{A}gt^{\bar{r}} = \{\bar{r}_A \mid A \subseteq \mathbb{A}gt, A \neq \varnothing\}$ is the set of "negative" deontic agents;
- $Q' = Q \cup \bigcup_{A\subseteq\mathbb{A}gt, A\neq\varnothing}(Q^{\mathsf{r_A}} \cup Q^{\bar{\mathsf{r}}_\mathsf{A}})$. We assume that Q and all $Q^{\mathsf{r_A}}, Q^{\bar{\mathsf{r}}_\mathsf{A}}$ are pairwise disjoint. Further we will be using the more general notation $S^{\mathsf{e}} = \{q^{\mathsf{e}} \mid q \in S\}$ for any $S \subseteq Q$ and proposition e;
- $\Pi' = \Pi \cup \{\mathsf{act}, ..., \mathsf{r_A}, ..., \bar{\mathsf{r}}_\mathsf{A}, ...\}$, and $\pi'(p) = \pi(p) \cup \bigcup_{A\subseteq\mathbb{A}gt}(\pi(p)^{\mathsf{r_A}} \cup \pi(p)^{\bar{\mathsf{r}}_\mathsf{A}})$ for every $p \in \Pi$. Moreover, $\pi'(\mathsf{act}) = Q$, $\pi'(\mathsf{r_A}) = Q^{\mathsf{r_A}}$, and $\pi'(\bar{\mathsf{r}}_\mathsf{A}) = Q^{\bar{\mathsf{r}}_\mathsf{A}}$;
- $d'_q(a) = d_q(a)$ for $a \in \mathbb{A}gt, q \in Q$: choices of the "real" agents in the original states do not change,
- $d'_q(r_A) = \{pass\} \cup img(q, \mathcal{R}_A)^{\mathsf{r_A}}$, and $d'_q(\bar{r}_A) = \{pass\} \cup (Q \setminus img(q, \mathcal{R}_A))^{\bar{\mathsf{r}}_\mathsf{A}}$. Action $pass$ represents a deontic agent's choice to remain passive and let other agents choose the next state. Note that other actions of deontic agents are simply labeled by the names of deontic states they point to;
- $Act' = Act \cup \bigcup_{q\in Q, A\subseteq\mathbb{A}gt}(d'_q(r_A) \cup d'_q(\bar{r}_A))$;
- the new transition function for $q \in Q$ is defined as follows (we put the choices from deontic agents in any predefined order):

$$\delta'(q, \alpha_{a_1}, ..., \alpha_{a_k}, ..., \alpha_r, ...) = \begin{cases} \delta(q, \alpha_{a_1}, ..., \alpha_{a_k}) & \text{if all } \alpha_r = pass \\ \alpha_r & \text{if } r \text{ is the first active (positive or negative) deontic agent} \end{cases}$$

- the choices and transitions for the new states are exactly the same: $d'(q^{\mathsf{r_A}}, a) = d'(q^{\bar{\mathsf{r}}_\mathsf{A}}, a) = d'(q, a)$, and $\delta'(q^{\mathsf{r_A}}, \alpha_{a_1}, ..., \alpha_{r_A}, ...) = \delta'(q^{\bar{\mathsf{r}}_\mathsf{A}}, \alpha_{a_1}, ..., \alpha_{r_A}, ...) = \delta'(q, \alpha_{a_1}, ..., \alpha_{a_k}, ..., \alpha_{r_A}, ...)$ for every $q \in Q, a \in \mathbb{A}gt', \alpha_a \in d'(q, a)$.

[3] We can check the last requirement by testing whether the transition leads to a deontic state of $r_{\mathbb{A}gt}$ (proposition $\mathsf{r}_{\mathbb{A}gt}$). It can happen only if all other relevant deontic agents choose action $pass$.

Now, we define a translation of formulae from DATL to ATL corresponding to the above described interpretation of DATL models into ATL models:

$$
\begin{array}{rcl}
tr(p) & = & p, \qquad \text{for } p \in \Pi \\
tr(\neg\varphi) & = & \neg tr(\varphi) \\
tr(\varphi \vee \psi) & = & tr(\varphi) \vee tr(\psi) \\
tr(\langle\langle A\rangle\rangle \bigcirc \varphi) & = & \langle\langle A \cup \mathbb{A}\mathrm{gt}^{r} \cup \mathbb{A}\mathrm{gt}^{\bar{r}}\rangle\rangle \bigcirc (\mathsf{act} \wedge tr(\varphi)) \\
tr(\langle\langle A\rangle\rangle \Box \varphi) & = & tr(\varphi) \wedge \langle\langle A \cup \mathbb{A}\mathrm{gt}^{r} \cup \mathbb{A}\mathrm{gt}^{\bar{r}}\rangle\rangle \bigcirc \langle\langle A \cup \mathbb{A}\mathrm{gt}^{r} \cup \mathbb{A}\mathrm{gt}^{\bar{r}}\rangle\rangle \Box \\
 & & (\mathsf{act} \wedge tr(\varphi)) \\
tr(\langle\langle A\rangle\rangle \varphi \mathcal{U} \psi) & = & tr(\psi) \vee (tr(\varphi) \wedge \langle\langle A \cup \mathbb{A}\mathrm{gt}^{r} \cup \mathbb{A}\mathrm{gt}^{\bar{r}}\rangle\rangle \bigcirc \langle\langle A \cup \mathbb{A}\mathrm{gt}^{r} \cup \mathbb{A}\mathrm{gt}^{\bar{r}}\rangle\rangle \\
 & & (\mathsf{act} \wedge tr(\varphi)) \mathcal{U} (\mathsf{act} \wedge tr(\psi))) \\
tr(\mathcal{O}_A \varphi) & = & \neg\langle\langle \mathbb{A}\mathrm{gt}^{r} \cup \mathbb{A}\mathrm{gt}^{\bar{r}}\rangle\rangle \bigcirc (\mathsf{r_A} \wedge \neg tr(\varphi)) \\
tr(\mathcal{UP}_A \varphi) & = & \neg\langle\langle \mathbb{A}\mathrm{gt}^{r} \cup \mathbb{A}\mathrm{gt}^{\bar{r}}\rangle\rangle \bigcirc (\bar{\mathsf{r}}_\mathsf{A} \wedge tr(\varphi)).
\end{array}
$$

Proposition 5.3 *For every* DATL *formula φ, model M, and a state $q \in Q$, we have $M, q \models \varphi$ iff $M^{\mathrm{ATL}}, q \models tr(\varphi)$.*

Proposition 5.4 *For every* DATL *formula φ, model M, and "action" state $q \in Q$, we have $M^{\mathrm{ATL}}, q \models tr(\varphi)$ iff $M^{\mathrm{ATL}}, q^{\mathsf{e}} \models tr(\varphi)$ for every $\mathsf{e} \in \Pi' \setminus \Pi$.*

Corollary 5.5 *For every* DATL *formula φ and model M, φ is satisfiable (resp. valid) in M iff $tr(\varphi)$ is satisfiable (resp. valid) in M^{ATL}.*

Note that the vocabulary (set of propositions Π) only increases linearly (and certainly remains finite). Moreover, for a specific DATL formula φ, we do not have to include all the deontic agents r_A and $\bar{r}_A$ in the model – only those for which $\mathcal{O}_A$ (or $\mathcal{UP}_A$, respectively) occurs in φ. Also, we need deontic states only for these coalitions A. The number of such coalitions is never greater than the complexity of φ. Let m be the cardinality of the "densest" modal accessibility relation – either deontic or temporal – in M, and l the complexity of φ. Then, the "optimized" transformation gives us a model with $m' = O(lm)$ transitions, while the new formula $tr(\varphi)$ is only linearly more complex than φ.[4] In consequence, we can use the ATL model checking algorithm from (Alur et al., 2002) for an efficient model checking of DATL formulae – the complexity of such process is $O(m'l') = O(ml^2)$.

Example 5.4 Let us consider again the deontic game structure from Figure 5.3. We construct a corresponding concurrent game structure, optimized for model checking of the DATL formula $\mathcal{O}_{\mathbb{A}\mathrm{gt}}(\neg\mathsf{pdead} \wedge \langle\langle k\rangle\rangle \bigcirc \neg\mathcal{O}_{\mathbb{A}\mathrm{gt}}\neg\mathsf{pdead})$: it is required that the Prime Minister is alive, but the killer is granted the ability to change this requirement. The result is shown in Figure 5.4. The translation of this formula is:

$$\neg\langle\langle r_{\mathbb{A}\mathrm{gt}}\rangle\rangle \bigcirc (\mathsf{r}_{\mathbb{A}\mathrm{gt}} \wedge \neg(\neg\mathsf{pdead} \wedge \langle\langle k, r_{\mathbb{A}\mathrm{gt}}\rangle\rangle \bigcirc (\mathsf{act} \wedge \neg\neg\langle\langle r_{\mathbb{A}\mathrm{gt}}\rangle\rangle \bigcirc (\mathsf{r}_{\mathbb{A}\mathrm{gt}} \wedge \neg\neg\mathsf{pdead}))))$$

which holds in states q_3 and $q_3^{\mathsf{r}_{\mathbb{A}\mathrm{gt}}}$ of the concurrent game structure. □

[4] The length of formulae may suffer an exponential blow-up; however, the number of *different subformulae* in the formula only increases linearly. This issue is discussed in more detail in Section 3.4.4.

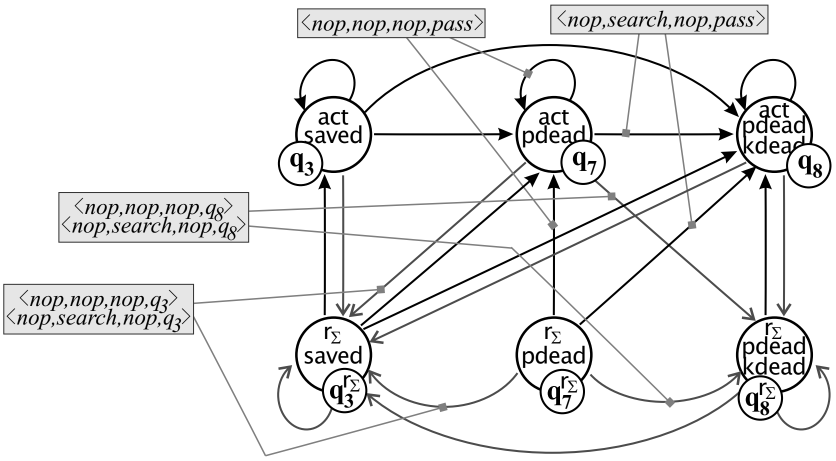


Figure 5.4: ATL interpretation for the deontic game structure from Figure 5.3

5.4.3 Planning to Achieve Deontic Goals

Having reduced model checking for DATL to ATL model checking, we can use the planning algorithm from Section 2.8 in order to generate plans that achieve goals that include deontic properties as well. The method closely resembles planning for epistemic goals from Section 3.4.5, and analogous remarks apply.

Example 5.5 Consider the local requirements from Example 5.3 again. Let M be the deontic concurrent game structure presented in that example, and let us assume that the dotted lines depict the obligations of James Bond (i.e., relation $\mathcal{R}_b$). If Bond is striving to be relieved from the tiresome duty of saving the world, then $\langle\langle b\rangle\rangle\Diamond\neg\mathcal{O}_b saved$ (more formally: $\langle\langle b\rangle\rangle\top\,\mathcal{U}\neg\mathcal{O}_b saved$) is the formula to be checked. Re-construction of the model yields concurrent game structure M^{ATL} from Figure 5.4 (only with deontic agent $r_{\mathbb{A}\mathrm{gt}}$ replaced with r_b), and the formula is translated to:

$$\langle\langle b, r_b\rangle\rangle \mathsf{act}\,\mathcal{U}(\mathsf{act} \wedge \langle\langle r_b\rangle\rangle \bigcirc (\mathsf{r_b} \wedge \neg\mathsf{saved})).$$

Now, executing $plan(\langle\langle b, r_b\rangle\rangle \mathsf{act}\,\mathcal{U}(\mathsf{act} \wedge \langle\langle r_b\rangle\rangle \bigcirc (\mathsf{r_b} \wedge \neg\mathsf{saved})))$ for M^{ATL} gives the following plan: $\{\langle q_7, -\rangle, \langle q_8, -\rangle\}$. In other words, the goal is already achieved in states q_7 and q_8, and impossible to achieve from q_3. Is there anybody else who can relieve Bond from duty? Yes, of course – the killer. We ask whether $\langle\langle k\rangle\rangle\Diamond\neg\mathcal{O}_b saved$, which translates as $\langle\langle k, r_b\rangle\rangle \mathsf{act}\,\mathcal{U}(\mathsf{act} \wedge \langle\langle r_b\rangle\rangle \bigcirc (\mathsf{r_b} \wedge \neg\mathsf{saved}))$, and the execution of $plan(\langle\langle k, r_b\rangle\rangle \mathsf{act}\,\mathcal{U}(\mathsf{act} \wedge \langle\langle r_b\rangle\rangle \bigcirc (\mathsf{r_b} \wedge \neg\mathsf{saved})))$ gives

$$\{\langle q_3, shoot\rangle, \langle q_7, -\rangle, \langle q_8, -\rangle\}.$$

Thus, if James Bond *really* wants to get rid of the obligation, then he must form a coalition with the killer (as $\langle\langle k\rangle\rangle\Diamond\neg\mathcal{O}_b saved$ implies $\langle\langle b, k\rangle\rangle\Diamond\neg\mathcal{O}_b saved$, and the same strategy can be used), or delegate the task to the killer in some other way. □

5.5 Conclusions

In this chapter, we have brought obligations and abilities of agents together, enabling one to reason about what coalitions should achieve, but also to formulate principles regarding who can maintain or reinstall the "correctness" of states of affairs or courses of action. We think the tractable model checking of DATL properties makes the approach attractive as a verification language for multi-agent systems that involve norms, obligations and/or requirements imposed on the system as a whole, or on individual agents. The language enables to express and verify specifications of agents' obligations, and confront them with abilities of the agents and their teams. But there is more to DATL than this: it makes also possible to reason about the temporal dynamics of the obligations, and to express the fact that someone can control the *requirements* themselves: formula $\langle\langle k\rangle\rangle\Diamond\neg\mathcal{O}_b saved$ from Example 5.5 illustrates the latter sort of ability. Last, but not least, we proposed an efficient planning algorithm that extends ATL-based planning with goals that involve deontic properties.

However, as stated repeatedly, it is at the same time a report of ideas rather than of a crystallized and final analysis. Few formal results were presented (it would be perhaps even fairer to say "suggested") for DATL in this chapter. Nevertheless, we believe that DATL is indeed a very attractive framework to incorporate abilities of agents and teams with deontic notions – and that there are many interesting features yet to be explored along this line of research. For instance, theoretical properties of DATL, and its relation to other existing systems that combine deontic and temporal/strategic perspective, wait to be investigated; in particular, a closer study of the relationship between DATL and the "Social Laws for ATL" approach seems worth conducting. Moreover, properties of a "guardian agent", "repairman" etc. are defined on the semantic level in the general case of local obligations – it can be interesting to try to express them in the object language as well, although it may require some redefinition of the semantics of deontic operators and/or cooperation modalities. Another line of research may refer to the other notion of obligation – obligations with respect to actions instead of states ("correct" actions rather than "good" states) – which can be confronted with agents' abilities as well. Finally, DATL can be extended with an epistemic dimension. Practical applications may include more realistic analysis of games, security analysis, trust management as well as requirements engineering.

Part II

Safer Decisions with Hierarchies of Models

Chapter 6

Bringing Adaptivity and Security Together

SYNOPSIS. *So far, various semantics for Alternating-time Temporal Logic were proved equivalent. Coalition Logic was shown to be subsumed by* ATL. *Their models and vocabulary were extended to handle obligations (requirements) and agents' beliefs under incomplete information – although the latter turned out not to be as easy and straightforward as it had seemed. What do we get from that? One can use multi-player game models to model environments inhabited by multiple agents, and the agents can employ* ATL *model checking to find infallible plans that satisfy their goals.*

But – what about fallible plans which are still good? What about exploiting the weaknesses of the opponents, or building trust and cooperation with other agents? How can we make our agents play relatively safe, and adapt to the changes of the dynamic environment at the same time? Multilevel modeling of the reality and multilevel decision making comes to rescue.

6.1 Introduction

This chapter presents the idea of hierarchical modeling of the reality. In many situations, a software agent can see several alternative models of his environment of action: differing in their structure, the way they have been obtained, and, most of all, the notions that underlie them. One model can include a profile of the user with whom the agent currently interacts, another one a stereotype or some "average user" model, a "best defense" model that assumes an adversary and powerful opponent etc. If the models are accurate beyond any doubt, then the agent should arguably use the most specific and detailed model while making his decisions; however, such a situation happens seldom in a dynamic environment. We propose that the agent can be better off using all the available models of the reality at the same time, and that the impact of a particular model should be proportional to its specificity *and* some evaluation of its accurateness and applicability to the actual case.

Chapters 2, 3, 4, and 5 show how multi-agent environments can be modeled via game-like scenarios, yielding multi-player game models and similar structures. The models can be extended to include agents' beliefs, desires, intentions, obligations, system requirements etc. It is not always as easy as it seems initially (cf. Chapter 4), but (assuming some care) one can imagine adding other notions to the scope of ATL-like models in a similar manner. What is very important, ATL model checking can be used as a planning algorithm for agents and their teams. However, ATL-based planning suffers from the inherent deficiency of game theory solutions: it looks for infallible plans, assuming in a sense that other agents (and even the "nature") plays *against* the planning agent. But what if the other agents are not necessarily adversary? Or if they are prone to make errors that could be exploited to reach the goals more easily? The agent should definitely be interested in learning some up-to-date knowledge about the environment and adapting his strategy accordingly. On the other hand, adaptivity can be risky if some opponent turns out to be powerful and adversary indeed. One of the advantages of using the multi-model decision making proposed here is that the agent can try to be (generally) adaptive and (relatively) secure at the same time.

The subsequent chapters address related issues: first, Chapter 7 reports research aimed at finding a good measure for the agent's self-evaluation of his actual beliefs; next, Chapter 8 shows how such adaptive-and-secure agents perform in a very simple setting. Chapter 8 presents also some examples how ATL models can be included in hierarchical modeling of the reality.

The preliminary idea of using a hierarchy of models was presented in (Jamroga, 2001b), and generalized in (Jamroga, 2002b). The chapter builds upon both papers.

6.2 Multilevel Modeling of Reality

A virtual agent lives in a world which consists of both virtual and "real" components. The world, together with the agent's own goals and capabilities, constitutes the reality the agent has to cope with. The agent interacts with the reality, trying to fulfill his (implicit or explicit) goals. Thus, it is good for the agent to learn some (implicit or explicit) model of the reality to adjust future actions to the predicted response of the environment.

6.2.1 Adaptivity vs. Security

An agent's knowledge about its environment can be either assumed ("pre-wired") or acquired through some kind of learning. The first approach dominates the classical game theory solutions – predefined, publicly known game trees, fixed payoffs, assumptions about players' rationality, and the maxmin equilibrium (von Neumann and Morgenstern, 1944), later generalized with the concept of non-transferable utility and Nash equilibrium for non-cooperative games (Nash, 1950) – define the best (or at least safest) choice in a normative way, assuming thus the optimal (or rather most dangerous) behavior of the "opponent". Recent modal logics of strategic ability, like ATL and CL, discussed extensively in Chapter 2, follow the same tradition. Their models generalize game trees, output of strategies is defined in a way analogous to maxmin, and verification of a formula in a given model generalizes minimax for zero-sum games (cf. Section 2.8). Similarly, "best defense models" for games involving uncertainty usually assume the opponent to play his best strategy, even to the extent of making him omniscient: both in game theory (Frank, 1996; Frank and Basin, 1998) and logics like ATEL or ATOL (cf. Chapters 3 and 4). Of course, omniscient and perfectly rational opponents are seldom met in the real world. Thus, such an assumption makes our agent over-cautious, although it protects the agent better in the case of meeting a powerful and strongly adversary enemy. An alternative solution was proposed in (Jamroga, 2001a): some boundaries of the possible opponent's knowledge have to be assumed (or learned), and within these boundaries we predict him to play average. The opponent can still be defined as omniscient, but it has to be done explicitly.

The machine learning approach emphasizes the importance of keeping an accurate and up-to-date model of the world. The agent can learn the policy of its adversary to exploit his weaknesses (Carmel and Markovitch, 1996; Sen and Arora, 1997; Sen and Weiss, 1999), to converge with dynamic, possibly indifferent environment (Sen and Sekaran, 1998; Sen and Weiss, 1999), or to learn trust and cooperation with other agents (Banerjee et al., 2000; Sen and Sekaran, 1998). The learning is accomplished mostly within the reinforcement learning regime (Kaelbling et al., 1996; Sen and Weiss, 1999). The goal of the agent is to maximize his numerical reward (payoff, utility) in the long run. Thus the decision making criterion is in most cases based on maximization of the expected payoff with respect to the agent's current knowledge about the environment of his action.

Remark 6.1 *Value systems (Pfeifer and Scheier, 1999) are sometimes used as an alternative for reinforcement learning. Instead of taking the raw reinforcement as the basis for his behavior, the agent tries to maximize the output of his own internal evaluation mechanism (his* value system*), which is only to some extent based on the external feedback. Thus, the agent is driven by a private system of preferences which may include biases towards specific situations and actions.*

In fact, value systems seem to provide a more general view to autonomous agents' learning than assuming immediate "internalization" of the external reinforcement. If an agent is autonomous, he should rather be supposed to reinterpret the feedback from the environment in his own, autonomous way. Of course, the difference is mainly philosophical. In the technical sense, the role of both reinforcement mechanisms and value systems is to provide the agent with numerical values that enable him to evaluate the utility of possible situations in some predefined sense (and, in consequence, also to evaluate his actions and strategies). We will refer to these values as payoffs or utilities throughout the rest of the thesis, and leave out the (no doubt interesting) issue where the utilities come from.

It is clear that an agent can benefit from learning up-to-date knowledge about his environment of action. However, some assumed "borderline" characteristic of the reality can still be very helpful when the agent's learned knowledge seems insufficient or cannot be sufficiently trusted.

6.2.2 Multiple Models of Reality

An agent may model his environment of action in many different ways, and in most cases it is up to the designer to decide which one will be maintained and used by the agent before he starts his "life". For example, the agent may perceive the environment as a unity – this approach pays off especially when the environment consists only of passive elements. Even in worlds (possibly) inhabited by other agents, it may be a good solution, as reported in (Sen and Sekaran, 1998) for block pushing agents using a very simple reinforcement learning scheme. However, if the agent can observe other agents' actions and distinguish them from changes of the environment itself, he may benefit from that in most cases. First, the agent is then able to monitor the state of the environment more precisely. Second, identifying separate (active) entities in the neighborhood creates a potential for dialogue in the broadest sense, as every agent-to-agent interaction may be seen as an instance of multimodal communication. Agents can be classified with respect to the way they model their environment of action in the following manner (Vidal and Durfee, 1998; Sen and Weiss, 1999):

- *0-level agent* is an agent who models the environment as a unity, i.e. he does not keep separate models of other agents;

- *1-level agent* is an agent who maintains and uses explicit models of other agents. In order to cut down the conceptual loop, the other agents are modeled as *0-level agents*;

- *2-level agent* is an agent who models other agents as 1-level agents;

- *k-level agent* is an agent who models other agents as $k - 1$-level agents.

In this chapter, we propose that the agent may be better off keeping several alternative models of the reality at the same time, and switching to the most appropriate at the very moment. Most notably, adaptive and normative models can be combined; ideally, the agent should base his decisions on the knowledge he has learned if the knowledge is trustworthy, and opt for "safe play" (e.g. maxmin) otherwise. Also, the concept of belief hierarchy presented here may enable using the content-based knowledge (individual user profiles) and the collaborative models (stereotypes) at the same time, especially for quantitative beliefs (Zukerman and Albrecht, 2001; Kobsa, 1993). However, we do not pursue the last idea within this thesis.

Similar intuition underlies a number of recent results: an adaptive news agent that keeps two complementary models of the user (long-term preferences + short-term ones) (Billsus and Pazzani, 1999), a system that uses alternative Markov models for predicting users' requests on a WWW server (Zukerman et al., 1999), etc. In both cases hybrid models are presented that perform better than any of the original models alone. Finally, some papers propose multilevel learning in order to learn user's interest that can possibly drift and recur (Koychev, 2001; Widmer, 1997).[1]

[1] I would like to thank Ingrid Breymann for her literature overview I dared to use.

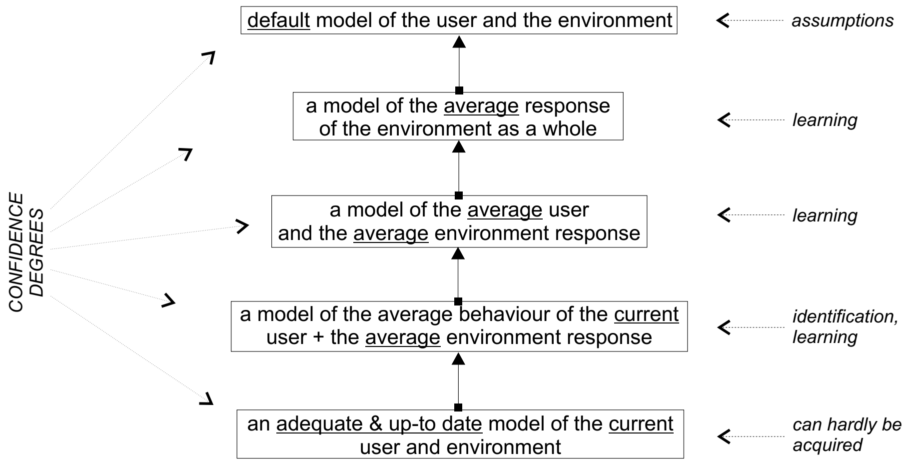


Figure 6.1: Example belief structure for an e-commerce agent

6.2.3 Modeling Dialogue Environment for e-Commerce Agents

The ideas, presented in this chapter, emerged from considerations about agents in business-to-consumer e-commerce. When a software agent has several communication modalities at hand (written text, speech, graphics, animation, video sequences, music samples etc.) – which of the modalities is optimal to present the information the agent is intended to provide? Or, putting it in a more general way, how can the e-commerce agent adjust to preferences of the so-called "user" (which is in fact a label for an ever-revolving cast of characters from the external reality)? The setting is not as adversary as it may seem at the first glance: every agent is interested in establishing successful communication, and in efficient information exchange – although the agents do not necessarily want to exchange *the same* information. Still, communication is usually a cooperative activity: for instance, the issue of making the communication fast, clear, attractive etc. is vital for all the agents being involved. On the other hand, the business agent must be cautious enough not to be cheated or manoeuvred into unprofitable contracts by a number of consumers in a long run.

An e-commerce agent should obviously be interested in possessing a perfectly adequate model of the environment. It may include the current user's preferences, his strategy, predicted future actions etc. However, such a model can hardly be acquired: the user may dynamically change his profile or even try to cheat the agent about his preferences, strategy or his identity. The agent can only try to build up some model of the average behavior/preferences presented so far by this particular user – if he is able to recognize the user in the crowd of all potential interlocutors. Some Internet agents try to identify the user by the IP number of the computer used by the user at this moment or through the cookies mechanism, some other force users to log in and confirm the identity with a password. But even humans, using all the available senses to recognize the interlocutor, are often full of doubts and make mistakes, especially when a new person appears in the scope of interest. Thus, the virtual agent may need to use some model of the "average user". This model can be learned simultaneously with particular users' profiles.

In a blurred, highly dynamic environment, where the agent cannot easily distinguish actions of other agents from changes of the environment itself, some model of the entire reality may turn out to be most useful and trustworthy. And, finally, when we cannot trust anything we learned so far, we need some default assumptions about the nature of the reality to evaluate possible courses of action and choose among them. All the proposed levels of modeling are shown in Figure 6.1. The more specific the level of knowledge used by the agent, the more accurate his decisions can be. However, if the agent has little or no confidence in his lower-level (more specific) beliefs, he should turn to the higher-level (more general) ones.

6.2.4 Inside the Boxes and Behind the Arrows

One can imagine expressing the actual agent's beliefs using very different languages. Also, the learning may proceed along different learning methods and routines.

Let us first consider the qualitative approach. The representation of agents' beliefs may be based on any kind of logic. For instance, the whole hierarchy may be defined in a way similar to a default theory in default logic (Antoniou, 1999) – or rather a "multi-default" logic in this case. The beliefs on a certain level of specificity can be therefore represented with sets of axioms. If some important fact cannot be proven on the most specific level of beliefs, the agent turns to the "local defaults" level (one level up); if there is still no answer to be found, he tries "the defaults over defaults", etc. Thus, the confidence degrees are defined (implicitly) in a usual binary fashion of mathematical logics: either the agent *is* confident with some formula (if it can be proven), or he is not.

The hierarchy may be also defined in a subsumption-like architecture, with explicit activation or inhibition links (if the confidence for the actual level is too low, the upper level is triggered on). The knowledge on every level can be expressed with any kind of representation language, including formulae of first-order predicate logic, logic programs, semantic networks, formulae of complex multimodal logics like ATL, ATOL or BDI (discussed in Chapters 2, 3 and 4 of this thesis), or even non-monotonic reasoning languages – learned via belief revision systems, inductive logic programming, genetic algorithms etc. The links may be triggered with respect to logical constraints, fuzzy logic formulae or numerical variables. If the beliefs are expressed with a non-monotonic logic, for example, an activation link may be triggered for a couple of steps every time a belief revision is necessary on the particular level (the agent had to change his beliefs seriously, so for some time he cannot trust them).

Of course modeling the environment with *models* of the above logics instead of their formulae is a worthy alternative. Multi-player game models, alternating observational transition systems, BDI models etc. may obviously serve to describe the structure and the current state of the reality. It is worth pointing out that in such case – instead of *proving* that some required formula holds – we must check whether it holds in a particular state of *the specified model*. In consequence, we replace theorem proving with model checking, which usually reduces the computational complexity of the decision-making procedure. Examples of hierarchies of models that include concurrent game

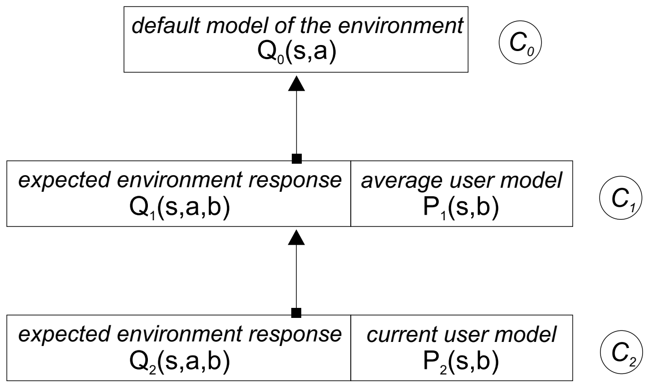


Figure 6.2: An example hierarchy for an e-commerce agent with probabilistic beliefs.

structures are presented in Section 8.3.

The subsumption-like architecture suits also defining multilevel beliefs within the quantitative approach. The beliefs can be gathered through Q-learning, genetic algorithms like BBA or PSP, statistical methods – yielding probability distributions, fuzzy sets or fuzzy measures, cluster models etc. They can also be accumulated in the form of Bayesian classifiers, Bayesian nets, neural networks and so on.

Example 6.1 An agent employing Q-learning to estimate the expected long-term reward (discounted over time) for his actions, with models of other agents as probability distributions over their possible choices (obtained via Bayesian updating), may produce the following hierarchy:

- $Q_0(s, a)$: the default expected reward for action a taken in state s;
- C_0: confidence the agent has in model Q_0. Note that we *assume* Q_0 to be correct by default, hence we can have no uncertainty about it as long as we keep the assumption ($C_0 = 1.0$);
- $Q_1(s, a, b)$: the average expected reward for action a in state s when the other agent executes action b;
- $P_1(s, b)$: the (estimated) probability that the average user takes action b in state s;
- C_1: confidence the agent has in Q_1 and P_1 being an accurate model of the average user and environment behavior;
- $Q_2(s, a, b)$: the expected reward for a in s against the current user playing b;
- $P_2(s, b)$: the (estimated) probability that the current user chooses b in s;
- C_2: confidence the agent has in Q_2 and P_2 being the model of the current behavior of the environment and the user;

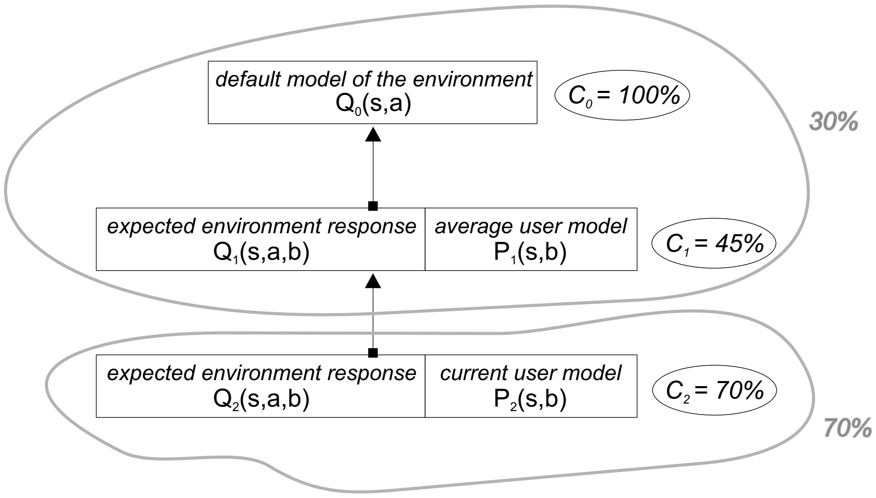


Figure 6.3: Combining probabilistic beliefs

The resulting hierarchy is shown in Figure 6.2. The agent should maximize his expected reward (depending on the belief level he uses at this moment), i.e.:

- he maximizes $E_0(s,a) = Q_0(s,a)$ if he bases his decision upon the default assumptions about the environment;
- he chooses a for which $E_1(s,a) = \sum_b Q_1(s,a,b)P_1(s,b)$ is maximal in the current state s if he uses the average user model;
- he chooses $a^* = \mathrm{argmax}_a\ E_2(s,a) = \mathrm{argmax}_a \sum_b Q_2(s,a,b)P_2(s,b)$ if he uses the model of the current user.

□

Note that within the quantitative approach the agent does not have to stick to one belief level only when evaluating possible alternatives. Suppose that confidence in a piece of knowledge (a model) is represented with a value $0 \leq C \leq 1$, with the intended interpretation that $C = 1$ denotes full confidence (the agent believes that the model is completely accurate), and $C = 0$ denoting complete distrust. Then the agent may use a linear combination of the evaluations as well, with the confidence values providing weights.

Example 6.2 If the agent trusts the most specific model in, say, 70% – the final evaluation should depend on the model in 70%, and the remaining 30% should be derived from the levels above. For the agent from Figure 6.3, the value to be maximized is:

$$\begin{aligned}E(s,a) &= C_2\,E_2(s,a) + (1-C_2)\,(C_1E_1(s,a) + (1-C_1)C_0E_0(s,a)) \\ &= 0.7\sum_b Q_2(s,a,b)P_2(s,b) + 0.135\sum_b Q_1(s,a,b)P_1(s,b) + 0.165\,Q_0(s,a).\end{aligned}$$

□

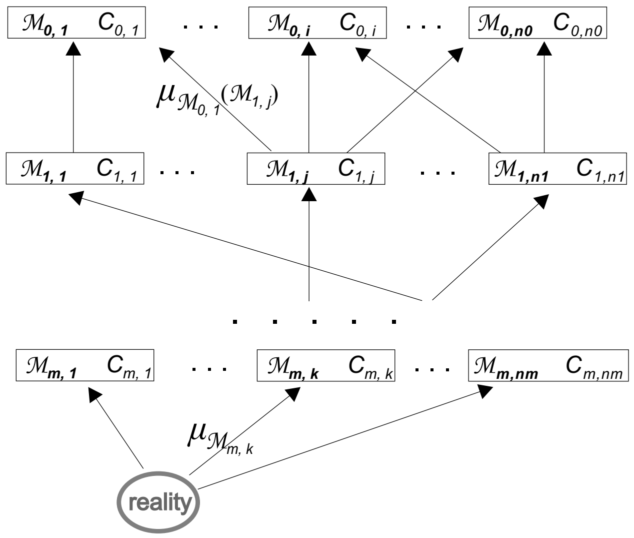


Figure 6.4: Generalized belief hierarchy for quantitative beliefs

In consequence, the decision is based on all the relevant models at the same time, although in different proportions – weighting the partial evaluations with the confidence the agent has in them.

6.3 Hierarchies of Quantitative Beliefs

The idea of hierarchical modeling of the reality has been presented on a few examples in the preceding section. In this section, we propose a more complex and general hierarchy of beliefs, in which several alternative models of the environment can be maintained and used on the same specificity level, including the level of default assumptions. The hierarchy enables multiple "inheritance" relation between concepts, and an arbitrary number of concept levels; moreover, the links between concepts can be also assigned numerical labels that describe their "strength".

6.3.1 Definitions

Let us assume that the agent's beliefs are quantitative in the sense that they imply some numerical evaluation of every action at hand, and that the decision making process can be based on the current evaluation values.

Definition 6.1 (Hierarchy of beliefs) *A* hierarchy of beliefs *is a directed acyclic graph, in which every node includes a model $\mathcal{M}$ of the agent's environment of action (the model can be also seen as a* concept *or a* notion *of the environment). A real number $C_{i,j}$, called* confidence value *is attached to node $\mathcal{M}_{i,j}$, and is meant to represent the agent's current degree of trust that $\mathcal{M}_{i,j}$ models the environment accurately. Every* *edge is labeled with a real number called a* membership value μ*, reflecting to what extent the lower-level model is a specialization of the higher-level one. The hierarchy is depicted in Figure 6.4.*

A few remarks may help to clarify the intuitions behind the hierarchy.

- The models are (partially) ordered with respect to their specificity. Top nodes (roots) describe the default assumptions about the behavior of the environment, other agents, users of the system etc. Bottom nodes (leaves) refer to the most specific available models: for instance, separate profiles for users, and models of how the environment reacts during interaction with each user respectively.

- Several alternative models on the same level refer to several possible classification decisions (with the same degree of specificity). For example, many user profiles can be kept at the bottom level; using a particular one depends on identification of the current user. Several competing models of the "average environment response" may refer to different types of the environment. In a sense, they can represent various *stereotypes* of the environment and/or users.

- Each model is underlied by some notion of the reality (its structure, operational characteristics etc.). The vertical links between models refer to the subset/membership relation between the notions, in a way similar to semantic networks (Russel and Norvig, 1995). For instance, a user of an e-banking system can be classified as an "honest customer" or a "dishonest" one (cf. Example 6.3). Such a classification depends usually on the actual evidence, and therefore implies some degree of uncertainty. Thus, the links are weighted with membership values that indicate to what extent we believe that the lower-level notion is a specific case of the higher-level notion. In consequence, $\mu_{\mathcal{M}}$ forms the characteristic function of a fuzzy set (Klir and Folger, 1988) that is supposed to model the notion behind $\mathcal{M}$.

- The membership values appear also below the most specific level of notions: the current "reality" can be classified as an instance of a particular notion only with some degree of certainty.

- Since the fuzzy nature of the relationships between notions is represented with the membership values μ, a confidence value C refers only to the agent's certainty that *the model in question describes the notion in question in an appropriate way*.

- The direction of arcs in the hierarchy is somewhat arbitrary, and reflects the intuition that we should start with the most specific model of the reality, and look for a more abstract one only when this one fails. On the other hand, we follow the tradition of placing the most abstract entries at the top, and most specific ones at the bottom of the hierarchy. Thus, the successors of a node are the nodes one level *up*.

- The root of the tree refers to the *real* state of affairs, and it is shown at the bottom of the graph. The fuzzy sets $\mathcal{M}_{m,1}, ..., \mathcal{M}_{m,n_m}$ have therefore only one member

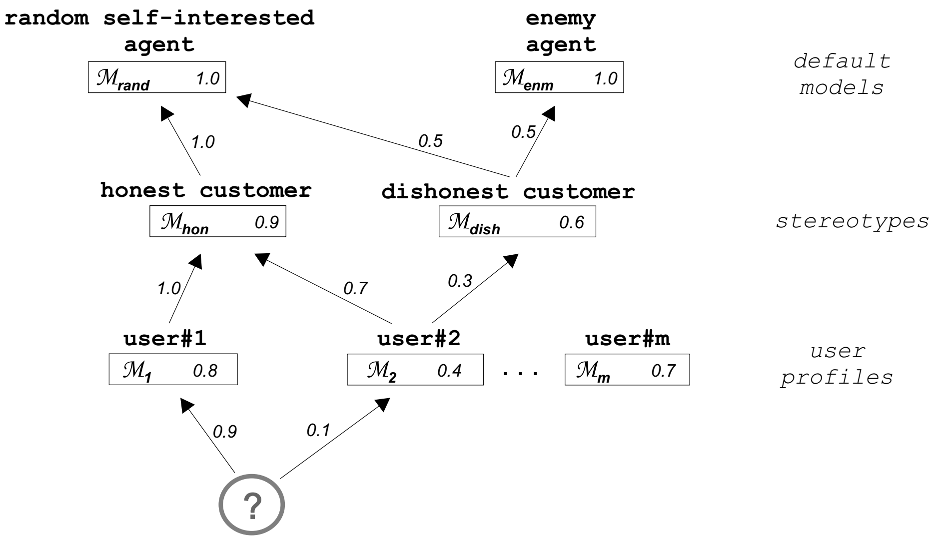


Figure 6.5: An example: two default models, two stereotypes, m user profiles

(the reality, that is), and the membership values $\mu_{\mathcal{M}_{m,1}}, ..., \mu_{\mathcal{M}_{m,n_m}}$ show how we interpret the observed data on the most specific level of knowledge.

Example 6.3 An example belief hierarchy for a banking agent is shown in Figure 6.5. It is assumed that the response from the environment is constant, deterministic and known beforehand, given the state of the system and the decisions from both agents (the e-banking agent and the current user). In consequence, building a model of the reality boils down to the task of *user modeling*.

The knowledge base includes m different user profiles. Two stereotypes: an "honest user" model, and a "dishonest user" model can be employed if there is substantial uncertainty about the profile of the current user. There are also two sets of default assumptions, describing an ordinary self-interested agent, and an enemy agent. □

Definition 6.2 (Multi-model evaluation of actions) *Let $succ(\mathcal{M})$ denote the set of all the successors of node $\mathcal{M}$, i.e. all the models exactly one level up. The (multi-model) evaluation of action a, starting from model $\mathcal{M}$, can be defined recursively:*

$$E(\mathcal{M}, a) \quad = \quad C_{\mathcal{M}} \cdot eval(\mathcal{M}, a) \; + \; (1 - C_{\mathcal{M}}) \sum_{\mathcal{M}' \in succ(\mathcal{M})} \mu_{\mathcal{M}'}(\mathcal{M}) \cdot E(\mathcal{M}', a)$$

where $eval(\mathcal{M}, a)$ is a numerical evaluation of a with respect to model $\mathcal{M}$ only – expected payoff estimation, for instance. Let $Spec$ denote the "bottom" nodes, i.e. the set of the most specific models. The final evaluation of alternative decisions can be now calculated as:

$$E(a) \quad = \quad \sum_{\mathcal{M} \in Spec} \mu_{\mathcal{M}} \cdot E(\mathcal{M}, a).$$

Example 6.4 The multi-model evaluation of action a, based on the hierarchy of models from Figure 6.5 (Example 6.3) is calculated as follows:

$$\begin{aligned}
E(a) &= 0.9\,E(\mathcal{M}_1,a) + 0.1\,E(\mathcal{M}_2,a) \\
&= 0.9\,(0.8\,eval(\mathcal{M}_1,a) + 0.2\cdot 1.0\,E(\mathcal{M}_{hon},a)) + \\
&\quad 0.1\,(0.4\,eval(\mathcal{M}_2,a) + 0.6\,(0.7\,E(\mathcal{M}_{hon},a) + 0.3\,E(\mathcal{M}_{dish},a))) \\
&= 0.72\,eval(\mathcal{M}_1,a) + 0.18\,(0.9\,eval(\mathcal{M}_{hon},a) + 0.1\,E(\mathcal{M}_{rand},a)) + \\
&\quad 0.04\,eval(\mathcal{M}_2,a) + 0.042\,(0.9\,eval(\mathcal{M}_{hon},a) + 0.1\,E(\mathcal{M}_{rand},a)) + \\
&\quad 0.018\,(0.6\,eval(\mathcal{M}_{dish},a) + 0.4\,(0.5\,E(\mathcal{M}_{rand},a) + 0.5\,E(\mathcal{M}_{enm},a))) \\
&= 0.72\,eval(\mathcal{M}_1,a) + 0.04\,eval(\mathcal{M}_2,a) + 0.1998\,eval(\mathcal{M}_{hon},a) + \\
&\quad 0.0108\,eval(\mathcal{M}_{dish},a) + 0.0582\,eval(\mathcal{M}_{rand},a) + \\
&\quad 0.036\,eval(\mathcal{M}_{enm},a).
\end{aligned}$$

□

The weights should be nonnegative and sum up to 1 finally (Kyburg, 1988); to assure this, the following restrictions on the belief structure are suggested.

Definition 6.3 (Additional requirements on the hierarchy)

1. $0 \le C_{\mathcal{M}} \le 1$ *and* $0 \le \mu_{\mathcal{M}'}(\mathcal{M}) \le 1$ *for every node* $\mathcal{M}$ *and* $\mathcal{M}'$ *(because the values are used to represent uncertainty);*
2. $\sum_{\mathcal{M}' \in succ(\mathcal{M})} \mu_{\mathcal{M}'}(\mathcal{M}) = 1$ *and* $\sum_{\mathcal{M}' \in Spec} \mu_{\mathcal{M}'} = 1$ *(i.e. no relevant notions are omitted in the hierarchy, and the notions do not overlap);*
3. $C_{0,i} = 1$ *for every* i *(the agent is fully committed to his most general assumptions).*

Now when the agent is able to compute some rating for every action, he can use any well-established decision-making scheme – like choosing the action with the highest expected payoff.

6.3.2 Verification of the Idea

Some simulations were conducted to verify the idea of keeping and using multiple alternative models of the reality – the results are presented and discussed in Chapter 8. In this place, however, we would like to give a preliminary idea how these simulations looked like, and in what way the results suggest that using such hierarchies of beliefs can be useful.

In this chapter, we basically propose that an agent may build up and use more than one model of the reality. In order to make things as simple as possible, the experiments employ an agent who interacts with a user in a stateless, stationary and deterministic environment with publicly known characteristic. The agents interacts with one user at a time, and the identity of the current user is always known beyond doubt. The agent uses exactly two models of the environment at a moment: a *profile* of the current user,

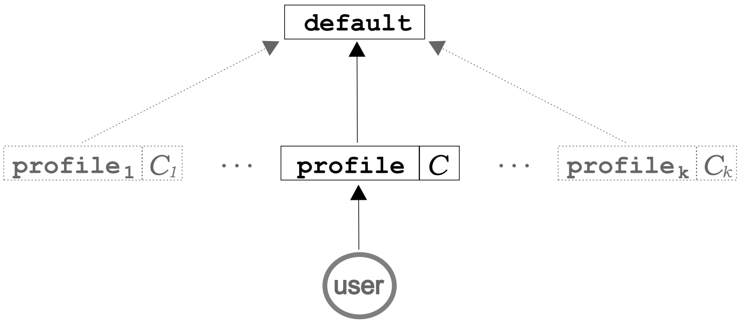


Figure 6.6: The simplest hierarchy: only two models of reality are relevant at a time

	accept	cheat	skip
risky offer	30	-100	1
normal offer	6	-20	1
safe offer	1.5	-1	1

Figure 6.7: Payoff table for the e-banking game

and a default "best opponent" model (see Figure 6.6). Moreover, the agent's utility function does not change throughout the game.

The simulations have been inspired by the following scenario: a software agent is designed to interact with users on behalf of an Internet banking service; he can make an offer to a user, and the user's response determines his output. The agent has 3 possible offers at hand: the "risky", "normal" and the "safe" offer, and the customer can respond with: "accept honestly", "cheat" or "skip". The complete table of payoffs for the game is given in Figure 6.7. The risky offer, for example, can prove very profitable when accepted honestly by the user, but the agent will lose proportionally more if the customer decides to cheat; as the user skips an offer, the bank still gains some profit from the advertisements etc.

Of course it is not essential that the agent is an e-banking broker. What is important is that he should learn users' profiles to approximate the actual preferences of each user. On the other hand, the agent has too much to lose to afford risky decisions when the identity of a user is unknown or the user is completely new to the system. To prevent this, he uses a default user model besides the profiles.

The banking agent is a 1-level agent, i.e. an agent that models other agents as 0-level stochastic agents. The user is simulated as a random static 0-level agent – in other words, his behavior can be described with a random probabilistic policy, and he does not change the policy throughout an interaction (a series of 100 rounds, consisting of an offer from the banking agent and a response from the user). To get rid of the exploration-exploitation tradeoff (Pfeifer and Scheier, 1999) we assume also that the user is rather simple-minded and his response does not depend on the actual offer being made: $p(cheat)$, $p(accept)$ and $p(skip)$ are the same regardless of the offer (if he is

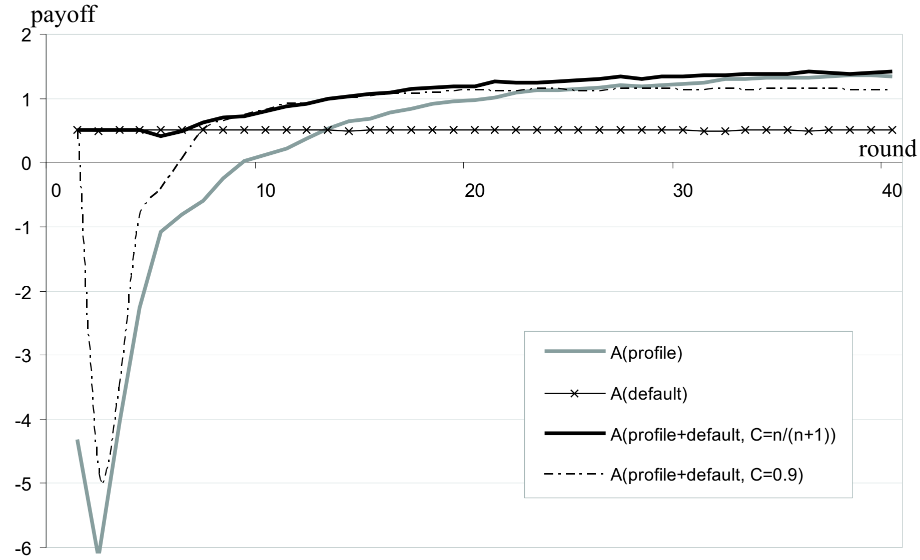


Figure 6.8: Two-level beliefs vs. using single model: average payoff per round

dishonest, he cheats for a small reward as well as a big one, for instance). The agent estimates the user's policy with a relative frequency distribution, counting the user's responses. The default model is defined in the game theory fashion: the user is assumed an enemy who always cheats. There is no uncertainty about the identity of the user – hence $\mu_{profile}(user) = 1$. As there is only one default model, $\mu_{default}(profile) = 1$; moreover, $C_{default} = 1$ (cf. Definition 6.3 pt. 3).

Remark 6.2 *One can suspect problems with obtaining appropriate confidence values. What we can do at least is to make sure that the confidence is low when the agent has collected few data so far, and that it is close to* 1 *when the data size is large. Some suggestions can be found in the the literature on statistical inference (Marshall and Spiegelhalter, 1999; Spiegelhalter et al., 1998) or higher-order uncertainty (Klir, 1999; Wang, 2001). The "variable confidence" agent defined below uses Wang's confidence:* $C = \frac{n}{n+1}$ *as the subsequent confidence values (Wang, 2001), where* n *is the number of observations (interaction rounds) completed so far. This – and other – confidence measures are studied in Chapter 7.*

The aim of the experiments was to compare the efficiency of such agent's behavior with the behavior of a standard learning agent – i.e. the agent who uses only a user profile when making his decisions. 1000000 independent random interactions (a sequence of 100 rounds each) have been simulated. Figure 6.8 shows the average payoff of the banking agent. 4 different agents were used: $A(\mathit{profile})$ denotes a single-model agent using only users' profiles, $A(\mathit{default})$ refers to another single-model agent who uses only the default "best defense" assumptions; $A(\mathit{profile} + \mathit{default}, C = 0.9)$ is an agent that employs both models with fixed confidence in the user profile $C = C_{profile} = 0.9$, and $A(\textit{profile} + \textit{default}, C = \frac{n}{n+1})$ denotes a double-model agent with variable confidence values.[2]

Remark 6.3 *Note that the single-model adaptive agent can be also interpreted as a special case of a double-model agent who always fully trusts his knowledge, i.e.* $A(\textit{profile}) = A(\textit{profile} + \textit{default}, C = 1)$*. Moreover, the single-model normative agent can be interpreted as a double-model agent who is never confident in his knowledge, i.e.* $A(\textit{default}) = A(\textit{profile} + \textit{default}, C = 0)$*.*

The output of the simulations shows that the banking agent can indeed benefit from using a default model together with the users' profiles in such setting. The last agent outperforms both single-model agents: he plays much safer in the first 25 rounds (when there is no sufficient data) and after that the payoffs are similar. Only the output of the first 40 rounds is presented on the chart to emphasize the part where the main differences lie. The results for rounds 41-100 were more or less the same.

6.3.3 Combining Evaluations vs. Combining Strategies

In Section 6.3.1, we proposed that agents can use multiple models of reality via combining *evaluations* of each possible strategy with respect to the available models. Another way is to combine best *strategies* directly – we can do it if we treat the strategies as mixed ones.

Definition 6.4 (von Neumann and Morgenstern, 1944) *Let* Σ *be a set of possible* pure *strategies of agent* a*, i.e. strategies that assign a deterministic choice to each game state. A* mixed strategy $s : \Sigma \rightarrow [0, 1]$ *is a probability distribution over* Σ*. We assume that* a *will draw his action at random from* Σ *with the probabilities defined by the distribution, if he commits to execute strategy* s*.*

If the set of pure strategies is finite $\Sigma = \{\sigma_1, ..., \sigma_n\}$*, then mixed strategies can be represented as vectors:* $s = [s(\sigma_1), ..., s(\sigma_n)]$*. Note that a pure strategy is a special kind of a mixed strategy:* $\sigma_1 = [1, 0, ..., 0]$*,* $\sigma_2 = [0, 1, ..., 0]$ *etc.*

A scalar multiplication and a sum of mixed strategies can be defined in a straightforward way. Let s and s' be mixed strategies over the same set of pure strategies Σ, and let r be a real number. Then:

- $(r \cdot s)(\sigma) = r \cdot s(\sigma)$;
- $(s + s')(\sigma) = s(\sigma) + s'(\sigma)$.

Note that not every linear combination of strategies must be a strategy itself.

Definition 6.5 (Multilevel combination of strategies) *Suppose that agent* a *uses the hierarchy of beliefs from Figure 6.4, and* $s_{\mathcal{M}}$ *denotes the (mixed) strategy of* a*, based*

[2] In fact, fixed $C = 0.5$ and 0.7 were also tried, but the results were virtually the same as for $C = 0$.

on model $\mathcal{M}$. Again, let $succ(\mathcal{M})$ denote the set of all the successors of node $\mathcal{M}$, and $Spec$ the set of all the most specific models. The multilevel strategy can be defined as:

$$S = \sum_{\mathcal{M} \in Spec} \mu_{\mathcal{M}} \cdot Strat(\mathcal{M})$$

$$Strat(\mathcal{M}) = C_{\mathcal{M}} \cdot s_{\mathcal{M}} + (1 - C_{\mathcal{M}}) \sum_{\mathcal{M}' \in succ(\mathcal{M})} \mu_{\mathcal{M}'}(\mathcal{M}) \cdot Strat(\mathcal{M}').$$

Proposition 6.4 *A multilevel combination of mixed strategies is a mixed strategy.*

Proof: First we prove that every $Strat(\mathcal{M})$ is a mixed strategy, i.e. it is a probability distribution over Σ. The proof follows by structural induction over the $\mathcal{M}$'s level of specificity. For $\mathcal{M}$ being a root of the hierarchy, we have: $Strat(\mathcal{M}) = s_{\mathcal{M}}$, qed. Suppose now that all $Strat(\mathcal{M}')$ are mixed strategies for models $\mathcal{M}'$ down to the level k. Take any model $\mathcal{M}$ from level $k + 1$. Then:

1. $Strat(\mathcal{M})(\sigma) \geq 0$ for every $\sigma \in \Sigma$, because it is a sum of nonnegative elements;
2. by the induction hypothesis, and the requirements from Definition 6.3:

$$\begin{aligned}\textstyle\sum_{\sigma \in \Sigma} Strat(\mathcal{M})(\sigma) &= C_{\mathcal{M}} \cdot \textstyle\sum_{\sigma \in \Sigma} s_{\mathcal{M}}(\sigma) + \\ &\quad (1 - C_{\mathcal{M}}) \textstyle\sum_{\mathcal{M}' \in succ(\mathcal{M})} \big(\mu_{\mathcal{M}'}(\mathcal{M}) \textstyle\sum_{\sigma \in \Sigma} Strat(\mathcal{M}')(\sigma)\big) \\ &= C_{\mathcal{M}} + (1 - C_{\mathcal{M}}) = 1.\end{aligned}$$

Thus, each $Strat(\mathcal{M})$ is a probability distribution over Σ, which implies that the multilevel combination $S = \sum_{\mathcal{M} \in Spec} \mu_{\mathcal{M}} \cdot Strat(\mathcal{M})$ must be a probability distribution over Σ, too. □

Example 6.5 Consider the agent from Section 6.3.2 who uses only the profile of the user (with confidence C computed after every step of interaction), and the default user model. If $s_{\textit{profile}}$ is the strategy that maximizes $eval_{\textit{profile}}(a)$, and $s_{\textit{default}}$ maximizes $eval_{\textit{default}}(a)$, then the resulting multilevel strategy is

$$S = C\ s_{\textit{profile}} + (1 - C)\ s_{\textit{default}}$$

If $s_{\textit{profile}}$ and $s_{\textit{default}}$ are pure strategies, the agent chooses the strategy based on the profile with probability C, and the default strategy otherwise. □

6.3.4 A Few Remarks before the Next Chapter Begins

The concept of the belief hierarchy is aimed to help a virtual agent to behave in a more robust, flexible and consistent way, especially when the agent cannot fully trust his beliefs or he can have several competing models of the reality. There is practically no restriction on the way the beliefs are expressed; also, the links between the levels can be defined in many ways. The experiments showed that an autonomous agent can get more payoff when using multiple models of the environment rather than just one model. The hierarchy requires only linear growth of computational power on the agent's part (with respect to the number of models being used), and the particular models can be constructed and updated in parallel since they are independent by definition – they only share the input data.

Do We Really Need to Keep Multiple Models of Reality?

The experiments were primarily designed to be simple: the user was static (so the confidence could be assumed an increasing function of the data set size), and the agent was using only two alternative models, one of them fixed and both having the same structure. Thus, the agent could arguably use Bayesian updating, for instance (Kyburg, 1987), to integrate both sub-models in this very case (starting with the default model and updating it sequentially as more user-specific evidence arrives). In consequence the agent would use a single model, and no confidence values would be necessary. However, things are not always like this. If the user is not static, his behavior may become suspect from time to time, so the agent can be better off turning back to the default model to some extent – but it does not seem clever to require that he abandons *all* the knowledge gathered so far, and starts the learning process from the scratch again. If both models are evolving, the agent must keep them anyway to proceed with the updates. Last but not least, the models may be built upon different structures (for example, the default model could be a simple Q-function with no probability at all) or they may represent different entities: conscious beliefs, unconscious beliefs, reflexes – and then it is not clear how they can be integrated at all.

It is worth noting that in the course of the simulations the agent did gain some additional profit when incorporating the "best defense" model against 0-level *random* agents. In other words, the agent benefited from assuming adversary play from an opponent who was *not* adversary by any means. More experiments, against other types of opponents, are presented and discussed in Chapter 8.

Learning to Learn

Another interesting thing we can do is to treat the confidence values as parts of the respective models. Now, the learning might also refer directly to the confidence degrees of various pieces of knowledge. For instance, the agent's value system may promote belief states with high confidence values on the most specific levels (providing the agent with positively greater rewards in such states), and thus motivating the agent to explore his environment. This might help to overcome the designer's problem of the exploration-exploitation tradeoff in a way: Instead of having the exploration routine predefined by the designer, the agent would be interested in learning the right proportions between the exploration and exploitation actions by himself. Thus, the designer may make the agent *learn to learn* without procedurally forcing the agent to explore the environment via the learning algorithm. This idea may be worth studying in the future; however, it is not investigated further in this thesis.

To Trust, or not to Trust

Trust plays an extremely important role in the human society. There are lots of social rules – both explicit and implicit – that humans are assumed to obey. Nobody claims that all the people do necessarily follow the rules, but most of the time we act as if we believed so. For instance, when you drive on a motorway, you assume that no one is going to drive on your lane from the opposite direction, that no child will run suddenly in front of your car, that every driver on the road knows how to drive a car, nobody is doped, and there is no madman or manic killer in the cars around you. Your trust is limited, of course, and you know such things happen, but if you wanted to take all these possibilities seriously you would never drive at more than 20 kilometers per hour.

However, if you have a weird feeling that other drivers act somehow strange, you will probably reduce your speed *before* you find out what is really going on. Pushing your trust too far may prove very dangerous. It does not mean that you change your general believes about drivers and highways instantly – when you are back on a motorway in a couple of days, you will accept the same assumptions more or less again – but at this moment *something is not as it ought to be*, so you "suspend" the beliefs now, and you turn to some more fundamental ones instead.

The example illustrates one interesting feature of human behavior. If you have some "emergency procedure" available at hand, and you have got enough self-confidence that you can recognize an emergency situation, then you can easier and safer put your trust in what you have learned about the world around you. The same should be applicable for artificial agents. The hierarchical modeling of the environment, proposed in this chapter, enables defining such emergency procedures. Several confidence measures, designed to detect situations when agents should rather suspend their trust, are studied in the next chapter.

Chapter 7

Looking for a Suitable Confidence Measure

SYNOPSIS. *Game-like logics and models were introduced, discussed, extended, analyzed, proposed for automatic multi-agent planning – but that is still not enough for most real-life applications. The logics show one side of the coin: the guaranteed profit one can make following his safest strategy – yet sometimes much more can be gained through exploiting (even dubious or vague) views of the environment of action. We have proposed hierarchies of beliefs and multi-model decision making to enable combining adaptivity and security to some extent. Can it help to resolve the dialectic tension between seizing opportunities and playing things completely safe? Might be. But first, we need to provide our agents with a way of assessing how good the "views of the environment" actually are.*

7.1 Introduction

An agent may benefit from keeping several alternative models of the reality in certain situations – the point has been advocated in Chapter 6. If the agent is designed to interact with users, he can be obviously better off keeping the users' profiles to approximate the actual preferences of each user. However, when the identity of a user remains unknown or the user is completely new to the system, an average user model or a default model may be used instead. While a standard machine learning algorithm will assume some arbitrary initial model of such a user (via uniform or random distribution, for instance), it should be clear that such knowledge must not be trusted when it comes to decision making, since the model is not supported by any data so far. Moreover, users' preferences may evolve, and even worse: some users may assume someone else's identity (incidentally or on purpose). This calls for a kind of self-reflection on the agent's part: a confidence measure is needed to determine to which extent every piece of knowledge can be considered reliable. If we provide the agent with such a measure, he can base his decisions on the most reliable model, or use a linear combination of all the appropriate models.

In this chapter, several confidence measures are proposed for an agent, interacting with other agents (users) in a very simple environment. The agent is meant to employ a kind of meta-reasoning to determine the level of reliability of the possessed knowledge. The aim of the confidence measure is to represent meta-(un)certainty – thus the actual confidence values range from 0 (complete distrust) to 1 (full confidence). Some researchers from the probability theory community suggest that – to solve the problem – we should take the agent's *knowledge* as a random quantity, and use its variance as a clue (Pearl, 1987; Kyburg, 1988). The suggestion has been followed in Sections 7.2 and 7.3, with rather negative results. Another possibility is explored in Section 7.4. The measure is based on self-information loss function (or log-loss function), used widely in information theory and universal prediction (Merhav and Feder, 1998) – and the experiments prove the idea promising.

This chapter uses ideas and results already published in (Jamroga, 2002a), (Jamroga, 2003b) and (Jamroga, 2003a).

7.1.1 Why Should We Doubt Our Beliefs?

There are roughly two possible sources of doubt for a learning agent. First, the agent may have collected too little data. For instance, when the agent starts interaction with a completely new user, his knowledge about the user is virtually none. However, the knowledge is utilized in the same way by most algorithms, regardless of the number of learning steps that have been taken so far.

Next, the environment might have changed considerably, so the collected data do not reflect its current shape.

The knowledge produced by a learning algorithm is often no more than a working hypothesis. It is necessary for the agent that he can make his decisions; however, trusting the knowledge blindly implies some additional assumptions which are not true in most real-life situations. It is good for the agent to have some measure of uncertainty about his own knowledge – to minimize the risk of a decision, especially in the case when he has several alternative models to choose among or combine.

7.1.2 Related Research

Confidence has been recognized an important and useful notion within the Machine Learning community. It was successfully used in the areas of movement recognition (Wang et al., 1999), speech recognition (Mengusoglu and Ris, 2001; Williams and Renals, 1997) or in mobile user location (Lei et al., 1999) for instance. In most papers the term "confidence" or "confidence measure" refers to the probability that the agent's *decision* (e.g. a medical diagnosis, a classification of an image, a sequence of words assigned to a spoken text etc.) is right – i.e. it refers to the probability that a particular patient really suffers from pneumonia, that there is really a car in the middle of the picture, that the user really said "open sesame" etc. In this chapter the term "confidence" refers to a subjective property of the *beliefs* (probabilistic or not) themselves, i.e. this is our own *knowledge* in which we may be more or less confident. Thus, the confidence can be viewed as meta-knowledge or, more precisely, meta-uncertainty. Such a confidence measure is often seen as a result of some posterior verification that follows the process of knowledge acquisition or decision making (Hochreiter and Mozer, 2001). Also, the confidence measure may be based on the amount of data we have available (Wang, 2001) or the way the observed patterns evolve (Pearl, 1987; Kyburg, 1988).

The confidence measures proposed in this chapter come perhaps closest to the measure proposed by Wang (2001): $C_{Wang} = n/(n+k)$, where n is the amount of data and k is an arbitrary fixed number. For instance, $C_{Wang} = n/(n+1)$ for $k = 1$. It seems simple and rather ad hoc, but turns out to work surprisingly well (Wang, 2001; Jamroga, 2002a, 2003b).

The time flow and the resulting devaluation of the old data and/or knowledge have also been a focus of several papers. Kumar (1998) uses a confidence measure to improve a Q-learning based algorithm for adaptive network routing. The measure is very simple – the confidence in every Q-value which has not been updated in the last step is subject to "time decay": $C_{new}(x) = \lambda\, C_{old}(x)$, where $\lambda \in (0, 1)$ is the decay constant. A similar idea was introduced in (Koychev, 2000) to track the user's drifting interests effectively. There is no explicit confidence measure there, however; instead, a scheme for "forgetting" old observations by an agent is proposed. Moreover – in contrast to Kumar's decaying *knowledge* – these are rather *data* that become gradually forgotten.

Another perspective to the task of adapting to the user's dynamic behavior is offered by the research on time series prediction – especially the universal prediction, where a prediction does not necessarily have to be a simple estimation of the next observation, but it can be a complex structure (a probability assessment, a strategy etc.), and the real underlying structure (the "source") generating the events is assumed to be unknown (Merhav and Feder, 1998). The universal prediction methods focus on finding a good predictor, not on assessing *how* good it is, though. A way of transforming the log-loss values into confidence values is proposed in Section 7.4 – and the results seem to be promising.

7.2 Datasize-Based Confidence

This section is focused on the first source of the agent's uncertainty: how much confidence can he have in his knowledge when there is not enough data to support it? The problem is analyzed in a very simple setting: the agent is assumed to be a 1-level agent – i.e. an agent that models other agents as stochastic agents (Vidal and Durfee, 1998) – and the users are 0-level agents with probabilistic policies. The reinforcement is known beforehand for every decision of the agent, given a response from the user, and the domain of action is stateless (or at least the agent's perception does not let him distinguish between different states of the environment). The agent tries to estimate the actual policy of the user calculating a frequency distribution, which can be further used to find the decision with the maximal expected reward. The aim of the confidence is to represent meta-(un)certainty about the agent's knowledge, so when he has several alternative models available he can choose among them or combine their output. Thus, the actual confidence values should range from 0 (complete distrust) to 1 (full confidence).

7.2.1 Self-Confidence with Insufficient Data

It is often assumed that the (un)certainty an agent can have about his knowledge is nothing but a meta-probability or meta-likelihood – cf. (Draper, 1995) for instance. On the other hand, there are researchers who argue against it (Kyburg, 1988; Wang, 2001). This seems to reflect the assumption that the meta-uncertainty should refer to the usability of the model. Indeed, meta-probability is not very useful in this case: even if we know for sure that the model is slightly different from the reality (in consequence, its meta-probability is exactly 0), it *does* matter whether it is close to the real situation or not (Wang, 2001). This is also the perspective adopted in this section. In this respect, some authors propose approaches based on some notion of error or fitting obtained through a posterior verification of the model (Hochreiter and Mozer, 2001; Spiegelhalter et al., 1998; Marshall and Spiegelhalter, 1999). However, the disconfidence studied here is *a priori* not *a posteriori* by definition – therefore any posterior reasoning can do no good here. In consequence, purely practical solutions may be very useful and work surprisingly well in particular situations (Kumar, 1998; Wang, 2001).

It has been suggested that, when the model is a probability distribution, the agent's self-confidence may be defined using the variance of the distribution treated as a random quantity itself (Pearl, 1987; Kyburg, 1988). Thus, the confidence measures being proposed and studied in this section are based on the notion of aggregate variance of the estimator provided by the learning process.

7.2.2 Frequency Distributions with Decay

Assume an autonomous e-banking agent A who interacts with some other agent B (the "user") according to the scenario from Section 6.3.2. The interaction with the user is sequential and it consists of subsequent turns: first A chooses to proceed with an action a from a finite set $ActA$, then B replies with some $b \in ActB$, then A does $a' \in ActA$ and so on. Let $p(b)$ denote the current probability of agent B choosing action b in a predefined context. Usually we will assume that the context is determined by the latest action of A, i.e. that $p(b)$ denotes the probability of agent B choosing action b as a response to a particular A's action $a*$. However, $p(b)$ may as well denote the probability of B choosing b in response to $a*$ in state s (if the context includes states of the environment), or the probability of b in general (if our model of the environment is sparser) etc.

In other words, $p(b)$ denote the current stochastic policy of B in the given context. A tries to estimate the policy with a relative frequency distribution $\hat{p}$:

$$\begin{aligned}
\hat{p}(b) &\leftarrow \begin{cases} \frac{\hat{p}(b)N\cdot\lambda+1}{N\cdot\lambda+1} & \text{if } b \text{ is the user's response} \\ \frac{\hat{p}(b)N\cdot\lambda}{N\cdot\lambda+1} & \text{else} \end{cases} \\
N &\leftarrow N\cdot\lambda+1
\end{aligned}$$

where $\lambda \in [0,1]$ is the decay rate implementing the way A "forgets" older observations in favor of the more recent ones to model users that may change their preferences dynamically (Kumar, 1998; Koychev, 2000, 2001). N represents the data size after collecting n observations. Since the older observations are used only partially (the first one with weight λ^{n-1}, the second: λ^{n-2} etc.), the real quantity of data we use is

$$N = \sum_{i=1}^{n} \lambda^{n-i} = \begin{cases} \frac{1-\lambda^n}{1-\lambda} & \text{for } 0 < \lambda < 1 \\ n & \text{for } \lambda = 1 \end{cases}$$

The nil distribution $\mathbf{0}(b) = 0$ is used as the initial one. If the decay rate is allowed to vary then $N = \sum_{i=1}^{n} \prod_{j=i+1}^{n} \lambda_j$, where $\lambda_1, ..., \lambda_n$ denote the actual decay rates at the moments when the subsequent observations and updates were made.

Note that $\hat{p}(b)$ is basically a sample mean of a Bernoulli variable, although it is a *mean with decay*.

Definition 7.1 Mean with decay *of a sequence* $(X_{i=1,...,n}) = (X_1, ..., X_n)$*, weighted with a series of decay values* $\lambda_1, ..., \lambda_n$*, can be defined as:*

$$M_{\lambda_{1..n}}(X_{i=1,...,n}) = \frac{\sum_{i=1}^{n}(\prod_{j=i+1}^{n} \lambda_j) X_i}{\sum_{i=1}^{n} \prod_{j=i+1}^{n} \lambda_j} = \frac{\sum_{i=1}^{n}(\prod_{j=i+1}^{n} \lambda_j) X_i}{N_{\lambda_{1..n}}}$$

Proposition 7.1 *A frequency distribution with decay* $\hat{p}_n(b)$ *is a mean with decay of* $Resp_{i=1,...,n}(b)$*, i.e.:* $\hat{p}_n(b) = M_{\lambda_{1..n}}(Resp_{i=1,...,n}(b))$*, where*

$$Resp(b) = \begin{cases} 1 & \textit{if b is the user's response} \\ 0 & \textit{otherwise} \end{cases}$$

Note also that for $\lambda = 1$ we obtain an ordinary frequency distribution with no temporal decay. Moreover, M_λ has some standard properties of a mean (the proofs are straightforward):

Proposition 7.2

1. $M_{\lambda_{1..n}}(X + Y) = M_{\lambda_{1..n}}(X) + M_{\lambda=1..n}(Y)$
2. $M_{\lambda_{1..n}}(aX) = aM_{\lambda_{1..n}}(X)$
3. $\sum_b M_{\lambda_{1..n}}(p_{i=1..n}(b)) = 1$ *if* p_i *is a probability function.*

Remark 7.3 *The weights assigned to the data sequence satisfy the forgetting constraints from (Koychev, 2000).*

Remark 7.4 *Mean with decay can be computed incrementally:*

$$M_{\lambda_{1..n}}(X_{i=1,...,n}) = \frac{\lambda_n M_{\lambda_{1..n-1}}(X_{i=1,...,n-1}) + X_n}{\lambda_n N_{\lambda_{1..n-1}} + 1}$$

Thus, the agent must only remember the current values of $M_{\lambda_{1..n}}(X_{i=1,...,n})$ *and* $N_{\lambda_{1..n}}$ *to re-compute the mean when new data arrive.*

7.2.3 Binding the Variance of Sampling

Handbooks on statistics like (Berry and Lindgren, 1996) suggest a way to determine whether an amount of data is enough to estimate the population mean EX with a sample mean $\overline{X}$: we assume some acceptable error level ε and as soon as the sampling dispersion (standard deviation, for instance) gets below this value: $\sigma(\overline{X}) \leq \varepsilon$, we feel satisfied with the estimation itself. Since the real deviation value is usually hard to obtain, an upper bound or an estimation can be used instead.

If we want the "satisfaction measure" to be continuous, it seems natural that the satisfaction is full 1 when the condition holds for $\varepsilon = 0$, and it decreases towards 0 as the dispersion grows. It is proposed here that the confidence for a frequency distribution $\hat{p}$ can be somehow proportional to $1 - \sum_b disp(b)$, and the variance $var(\hat{p}(b))$ is used to express the dispersion $disp(b)$. The reason for choosing the variance is that $0 \leq \sum_b var(\hat{p}(b)) \leq 1$ in our case, while the same is not true for the standard deviation σ as well as the mean deviation $m.a.d.$

We assume that the old observations are appropriate only partially with respect to the (cumulative) data decay encountered so far. Let $n \geq 1$ be an arbitrary number. By the properties of the variance and given that $Resp_1(b), ..., Resp_n(b)$ represent a random sampling of the user's responses:

$$\begin{aligned} var(\hat{p}_n(b)) &= var\big(M_\lambda(Resp_{i=1..n}(b))\big) = \\ &= var\Big(\frac{\sum_{i=1}^n Resp_i(b)\lambda^{n-i}}{\sum_{i=1}^n \lambda^{n-i}}\Big) = \\ &= \frac{\sum_{i=1}^n var(Resp_i(b))\lambda^{2(n-i)}}{(\sum_{i=1}^n \lambda^{n-i})^2} \end{aligned}$$

The value of $var(Resp_i(b))$ is a population variance at the moment when the ith observation is being made. If $p_i(b)$ is the real probability of user responding with action b at that particular moment, then:

$$\begin{aligned} var(Resp_i(b)) &= p_i(b) - p_i^2(b) \\ \sum_b var(\hat{p}_n(b)) &= \frac{\sum_{i=1}^n \lambda^{2(n-i)}\big(\sum_b p_i(b) - \sum_b p_i^2(b)\big)}{(\sum_{i=1}^n \lambda^{n-i})^2} \end{aligned}$$

$\sum_b p_i^2(b)$ is minimal for the uniform distribution $p_i(b) = \frac{1}{|ActB|}$, so:

$$\sum_b var(\hat{p}_n(b)) \leq \frac{\sum_{i=1}^n \lambda^{2(n-i)}}{(\sum_{i=1}^n \lambda^{n-i})^2}(1 - \frac{1}{|ActB|})$$

Definition 7.2 *We define the measure Cbound as:*

$$\begin{aligned} \textit{Cbound} &= 1 - dispb, \quad \textit{where} \\ dispb &= \frac{\sum_{i=1}^n \lambda^{2(n-i)}}{(\sum_{i=1}^n \lambda^{n-i})^2}(1 - \frac{1}{|ActB|}) \end{aligned}$$

Now the confidence is never higher than it *should* be – the agent is playing it safe:

Proposition 7.5 $\mathit{Cbound} \leq 1 - \sum_b var(\hat{p}_n(b))$.

Note also that $dispb$ is a decreasing function of λ for $\lambda \in (0, 1]$, so its value is always between $(1 - \frac{1}{|ActB|})/n$ (the value for $\lambda = 1$), and $1 - \frac{1}{|ActB|}$ (which is $\lim_{\lambda \to 0} dispb$).

Corollary 7.6 $0 \leq \frac{1}{|ActB|} \leq \quad \mathit{Cbound} \quad \leq \frac{n-1}{n} + \frac{1}{n|ActB|} \leq 1$.

Definition 7.3 *In the more general case when λ is variable, the confidence can be defined as*

$$\mathit{Cbound} = 1 - \frac{SLsqr_n}{(SL_n)^2}(1 - \frac{1}{|ActB|}),$$

where $SLsqr_n$ and SL_n are computed incrementally:

$$\begin{aligned} SLsqr_n &= \sum_{i=1}^{n} \left(\prod_{j=i+1}^{n} \lambda_j \right)^2 = \lambda_n^2 SLsqr_{n-1} + 1 \\ SL_n &= \sum_{i=1}^{n} \prod_{j=i+1}^{n} \lambda_j = \lambda_n SL_{n-1} + 1 \end{aligned}$$

Note that:

$$\begin{aligned} \sum_b var(\hat{p}_n(b)) &= var\big(M_{\lambda=1..n}(Resp_{i=1..n}(b))\big) = \\ & \frac{\sum_b \sum_{i=1}^{n} (\prod_{j=i+1}^{n} \lambda_j)^2 var(Resp_i(b))}{(\sum_{i=1}^{n} \prod_{j=i+1}^{n} \lambda_j)^2} \leq \\ & \frac{\sum_{i=1}^{n} (\prod_{j=i+1}^{n} \lambda_j)^2}{(\sum_{i=1}^{n} \prod_{j=i+1}^{n} \lambda_j)^2}(1 - \frac{1}{|ActB|}) = 1 - \mathit{Cbound}. \end{aligned}$$

Corollary 7.7 $0 \leq \mathit{Cbound} \leq 1 - \sum_b var(\hat{p}_n(b))$.

7.2.4 Adjusting the Confidence Value

The value of $dispb$ proposed above can give some idea of the uncertainty the agent should have in $\hat{p}$. The most straightforward solution: $\mathit{Cbound} = 1 - dispb$ may not always work well for practical reasons, though. The agent can use a "magnifying glass" parameter $\mathfrak{m}$ to sharpen his judgment:

$$\mathit{Cbound} = (1 - dispb)^{\mathfrak{m}}.$$

Since different learning methods show different dynamics of knowledge evolution, $\mathfrak{m}$ offers the agent an opportunity to "tune" his confidence measure to the actual learning algorithm.

7.2.5 Forward-Oriented Distrust

In a perfect case we would be interested in the *real* variation of the sampling made so far – to have some clue about the expected (real) deviation from the estimation $\hat{p}_n$ obtained through the sampling. This value can be approached through its upper bound – as proposed in Section 7.2.3. Alternatively we can try to approximate the variability we may expect from our estimator in the future (possibly with temporal discount).

It is worth noting that insufficient data can be seen as generating "future oriented distrust": even if the agent's knowledge does not change much during the first few steps (e.g. the corresponding user's responses are identical) it may change fast in the very next moment. When the evidence is larger, the model of the reality being produced gets more stable and it can hardly be changed by a single observation. Let us assume that the learning algorithm is correct – i.e. the model converges to the true user characteristics as the number of input data increases.

Definition 7.4 *The agent can base his self-assessment on the possible future-oriented dispersion (possibly with a temporal discount* Λ*):*

$$\begin{aligned} Csize_\Lambda &= (1 - fdisp_\Lambda)^{\mathsf{m}} \\ fdisp_\Lambda &= \lim_{k\to\infty} E\, fdisp_\Lambda^k \;=\; \lim_{k\to\infty} E\big(\sum_b V_\Lambda(\hat{p}_{n+k}(b), ..., \hat{p}_n(b))\big) \end{aligned}$$

where $\hat{p}$ *is the agent's current model of the user, every* $\hat{p}_{n+i}, i = 1..k$ *is obtained from* $\hat{p}_{n+i-1}$ *through response* b_i^**, and the mean is taken over all the response sequences* $(b_1^*, ..., b_k^*)$*.*

Note that Λ is the decay rate for *knowledge*, and does not have to be the same as the observational decay rate λ.

Definition 7.5 *The sample variance with discount/decay can be defined in a natural way as:*

$$V_\Lambda(X) = M_\Lambda(X - M_\Lambda X)^2.$$

Proposition 7.8 *By properties of the mean with decay (Proposition 7.2):*

$$V_\Lambda(X) = M_\Lambda(X^2) - M_\Lambda^2(X).$$

The limit in Definition 7.4 can be approximated iteratively for the generalized frequency counting presented in Section 7.2.2, assuming uniform a priori likelihood for all possible sequences, and approximating the expected value through simple averaging.

Definition 7.6 *Let:*

$$\begin{aligned} avg_{(b_{i=1..k}^*)}\, fdisp_\Lambda^k &= \frac{1}{|ActB|^k} \sum_{(b_{i=1..k}^*)} fdisp_\Lambda^k \\ Mpsqr^k &= avg_{(b_{i=1..k}^*)} \sum_b M_\Lambda(\hat{p}_{n+k}^2(b), ..., \hat{p}_n^2(b)) \end{aligned}$$

$$\begin{aligned}
Msqr^k &= avg_{(b^*_{i=1..k})} \sum_b M^2_\Lambda(\hat{p}_{n+k}(b), ..., \hat{p}_n(b)) \\
MP^k &= avg_{(b^*_{i=1..k})} \sum_b \hat{p}_{n+k}(b) M_\Lambda(\hat{p}_{n+k}(b), ..., \hat{p}_n(b)) \\
Psqr^k &= avg_{(b^*_{i=1..k})} \sum_b \hat{p}^2_{n+k}(b)
\end{aligned}$$

Now, for $0 < \Lambda < 1$ being the *knowledge* decay rate, $\lambda = \lambda_n$ the current *observation* decay rate, $N = \sum_{i=1}^n \lambda_i$ being the current (decayed) data size, and $N_k = N\lambda^k + \sum_{i=0}^{k-1} \lambda^i$, we have:

$$\begin{aligned}
avg_{(b^*_{i=1..k})} fdisp^k_\Lambda &= Mpsqr^k - Msqr^k \\
Mpsqr^k &= \frac{1-\Lambda^k}{1-\Lambda^{k+1}} Mpsqr^{k-1} + \frac{(1-\Lambda)\Lambda^k}{1-\Lambda^{k+1}} Psqr^k \\
Msqr^k &= \frac{(1-\Lambda^k)^2}{(1-\Lambda^{k+1})^2} Msqr^{k-1} + \frac{2(1-\Lambda)\Lambda^k}{1-\Lambda^{k+1}} MP^k \\
&\quad - \frac{\Lambda^{2k}(1-\Lambda)^2}{(1-\Lambda^{k+1})^2} Psqr^k \\
MP^k &= \frac{(1-\Lambda^k)(N_k-1)}{(1-\Lambda^{k+1})N_k} MP^{k-1} + \\
&\quad \frac{(1-\Lambda)\Lambda^k}{1-\Lambda^{k+1}} Psqr^k + \frac{1-\Lambda^k}{|ActB|(1-\Lambda^{k+1})N_k} \\
Psqr^k &= \left(\frac{N_k-1}{N_k}\right)^2 Psqr^{k-1} + \frac{2(N_k-1)}{|ActB|N_k^2} + \frac{1}{N_k^2}
\end{aligned}$$

The resulting algorithm for iterative approximation of *fdisp* is shown in Figure 7.1.

Proposition 7.9 *The algorithm from Figure 7.1 is convergent.*

Proof: To prove the convergence of sequence $V^k = avg_{(b^*_{i=1..k})} fdisp^k_\Lambda$, we will find an (a_k) such that $|V^k - V^{k-1}| \le a_k$ for every k, and $\sum_{i=1}^k a_i$ forms a convergent series. Then the series $\sum_{i=1}^k (V^i - V^{i-1}) = V^k$ is also convergent. Note that:

$$\begin{aligned}
&|V^k - V^{k-1}| = |Mpsqr^k - Mpsqr^{k-1} + Msqr^{k-1} - Msqr^k| = \\
&= |(\frac{1-\Lambda^k}{1-\Lambda^{k+1}} - 1) Mpsqr^{k-1} + (1 - \frac{(1-\Lambda^k)^2}{(1-\Lambda^{k+1})^2}) Msqr^{k-1} \\
&+ \frac{(1-\Lambda)\Lambda^k}{1-\Lambda^{k+1}} Psqr^k - \frac{2(1-\Lambda)\Lambda^k}{1-\Lambda^{k+1}} MP^k + \frac{\Lambda^{2k}(1-\Lambda)^2}{(1-\Lambda^{k+1})^2} Psqr^k| \\
&\le |\frac{\Lambda^k(\Lambda-1)}{1-\Lambda^{k+1}} Mpsqr^{k-1}| + |\frac{\Lambda^k(2-2\Lambda-\Lambda^k+\Lambda^{k+2})}{(1-\Lambda^{k+1})^2} Msqr^{k-1}| \\
&+ |\frac{(1-\Lambda)\Lambda^k}{1-\Lambda^{k+1}} Psqr^k| + |\frac{2(1-\Lambda)\Lambda^k}{1-\Lambda^{k+1}} MP^k| + |\frac{\Lambda^{2k}(1-\Lambda)^2}{(1-\Lambda^{k+1})^2} Psqr^k|
\end{aligned}$$

fdisp ($\hat{p}$, Λ, λ, N, precision);

Iterative approximation of *fdisp*$_\Lambda$. Returns the approximate value of the average future-oriented temporally-discounted dispersion (with temporal discount $0 < \Lambda < 1$). λ is the current temporal decay rate on the observation level, N represents the number of observations collected so far that takes into account the cumulative time decay; $\hat{p}$ is the current model of the user.

```
Mpsqr, Msqr, MP, Psqr ← Σ_b p̂²(b);          /* initial values */
k ← 0;
V ← 0;
repeat
    V_old ← V;
    k ← k + 1;
    N ← Nλ + 1;
    Psqr ← ((N−1)/N)² Psqr + 2(N−1)/(|ActB| N²) + 1/N²;
    MP ← ((1−Λ^k)(N−1))/((1−Λ^{k+1})N) MP + ((1−Λ)Λ^k)/(1−Λ^{k+1}) Psqr + (1−Λ^k)/(|ActB|(1−Λ^{k+1})N);
    Msqr ← (1−Λ^k)²/(1−Λ^{k+1})² Msqr + (2(1−Λ)Λ^k)/(1−Λ^{k+1}) MP − (Λ^{2k}(1−Λ)²)/(1−Λ^{k+1})² Psqr;
    Mpsqr ← (1−Λ^k)/(1−Λ^{k+1}) Mpsqr + ((1−Λ)Λ^k)/(1−Λ^{k+1}) Psqr;
    V ← Mpsqr − Msqr;
until |V − V_old| ≤ precision;
return(V);
```

Figure 7.1: The algorithm for iterative approximation of *fdisp*.

$$\leq \frac{\Lambda^k(1-\Lambda)}{1-\Lambda^{k+1}} + \frac{\Lambda^k(1-\Lambda)(2-\Lambda^k(1+\Lambda))}{(1-\Lambda^{k+1})^2} + $$
$$+ \frac{\Lambda^k(1-\Lambda)}{1-\Lambda^{k+1}} + \frac{2\Lambda^k(1-\Lambda)}{1-\Lambda^{k+1}} + \frac{\Lambda^{2k}(1-\Lambda)^2}{(1-\Lambda^{k+1})^2}$$

because $0 \leq Mpsqr, Msqr, MP, Psqr \leq 1$ by Proposition 7.2 (point 3). Thus

$$\begin{aligned} |V^k - V^{k-1}| &\leq \frac{\Lambda^k(1-\Lambda)}{(1-\Lambda^{k+1})^2} + 2\frac{\Lambda^k(1-\Lambda)}{(1-\Lambda^{k+1})^2} + \frac{\Lambda^k(1-\Lambda)}{(1-\Lambda^{k+1})^2} \\ &\quad + 2\frac{\Lambda^k(1-\Lambda)}{(1-\Lambda^{k+1})^2} + \frac{\Lambda^k(1-\Lambda)}{(1-\Lambda^{k+1})^2} \leq \\ &\leq 7\frac{\Lambda^k(1-\Lambda)}{(1-\Lambda^{k+1})^2} \leq 7\frac{\Lambda^k(1-\Lambda)}{(1-\Lambda)^2} = \frac{7}{1-\Lambda}\Lambda^k. \end{aligned}$$

□

Note also that, for every k, we have $V^k \geq 0$ (because it is a sum of nonnegative elements). On the other hand, $V^k \leq 1$ (because $V^k = Mpsqr^k - Msqr^k$, and $0 \leq Mpsqr^k, Msqr^k \leq 1$).

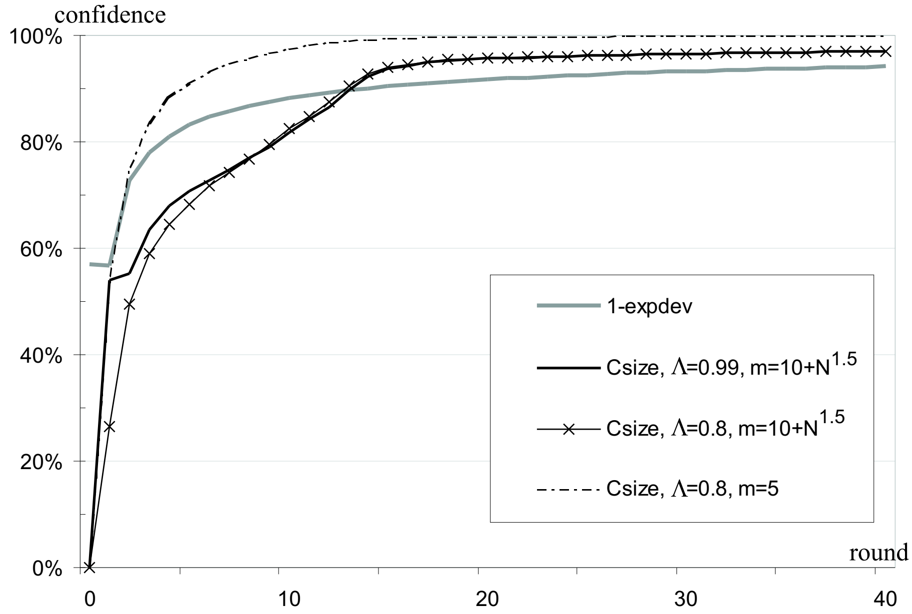


Figure 7.2: Confidence vs. accurateness: *Csize*

Corollary 7.10 $0 \leq Csize_\Lambda \leq 1$.

7.2.6 Simulations

The experiments were inspired by the e-banking scenario, depicted in Section 6.3.2. The banking agent can proceed with 3 different offers at each round, and the user has 3 different responses available. The table of payoffs for the game is shown in Figure 6.7. The agent estimates the user's policy with a relative frequency distribution, counting the user's responses; at the same time he computes a confidence value for the profile acquired so far. 1000000 independent interactions (a sequence of 100 rounds each) with a random user process have been simulated; the average results are presented in the following charts. Only the output of the first 40 rounds is presented in most charts to emphasize the part where the main differences lie. The results for rounds $41 - 100$ were more or less the same.

Figures 7.2, 7.3 and 7.4 show how the confidence values evolve for a static random user (a user with a random stochastic policy that does not change throughout the experiment). The banking agent uses plain frequency counting with no decay (i.e., the decay rate for observations is $\lambda = 1$). Figure 7.2 presents the characteristic of *Csize*, Figure 7.3 displays the flow of *Cbound*, and Figure 7.4 shows the characteristic of the Wang's confidence *Cwang* $= N/(N + k)$. The confidence values are compared against the expected absolute deviation of the learned profile from the real policy of the user: $expdev = \sum_b |\hat{p}(b) - p(b)| \cdot p(b)$, or rather the "accurateness" of the profile, i.e.

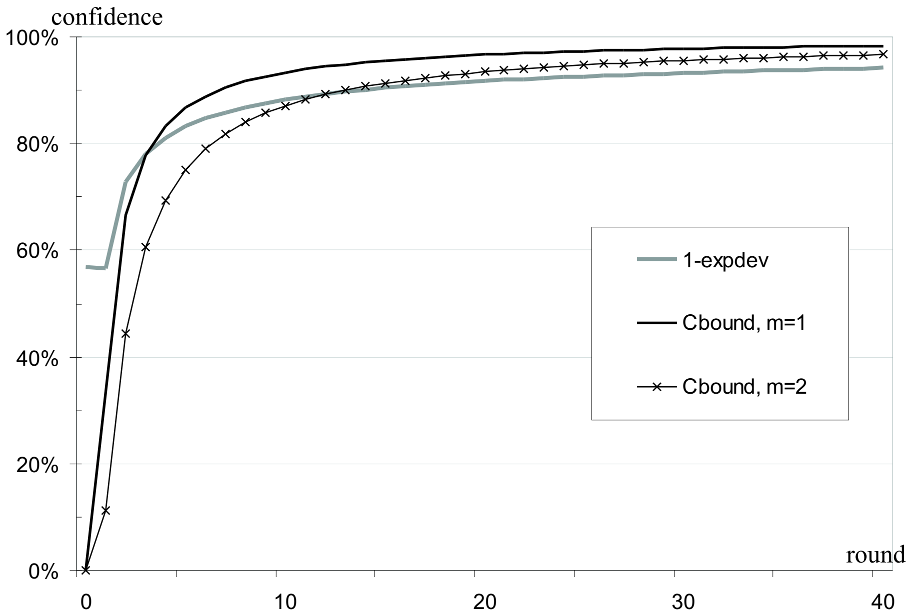


Figure 7.3: Confidence vs. accurateness: *Cbound*

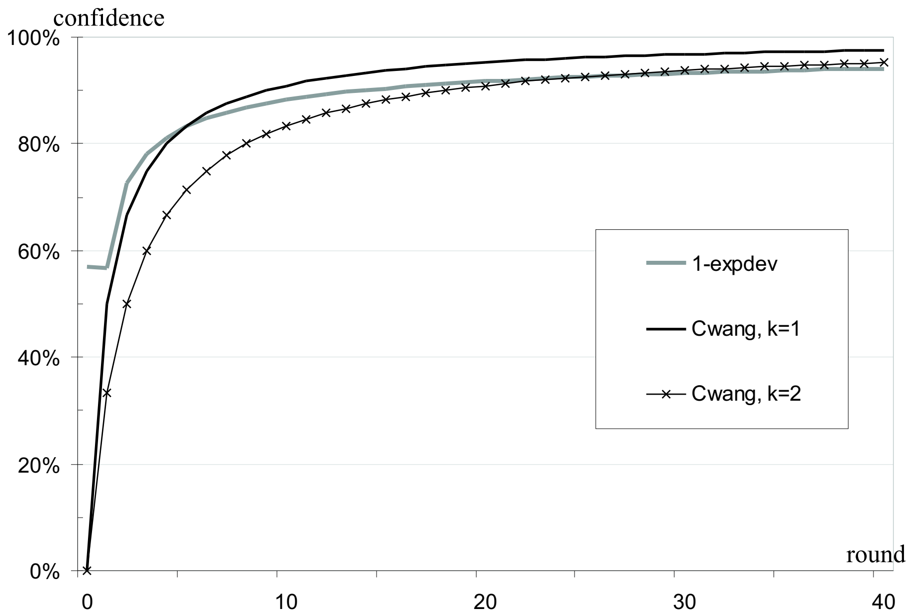


Figure 7.4: Wang's confidence for $k = 1$ and 2

$1 - expdev$.

All the presented measures increase as the number of collected data grows, which is a good thing. Obviously, a higher value of the "magnifying glass" parameter $\mathfrak{m}$ makes the confidence values increase slower – and in consequence the agent is more "hesitant" to accept the evidence coming from his observations (cf. Figures 7.2 and 7.3). Using a higher value of k in *Cwang* has the same effect (Figure 7.4). Note that the the measure based on the "forward-oriented distrust" (*Csize*) goes very fast to 1, even for a relatively high "magnifying glass" ($\mathfrak{m} = 5$). Too fast, in fact: the performance of agents using *Csize* with fixed $\mathfrak{m}$ turned out to be almost the same as for agents using only the user's profile and no multilevel decision making at all. Looking for a successful set of parameters we tried a more complicated magnifying scheme eventually: variable $\mathfrak{m} = 10 + N^{1.5}$. The resulting confidence is relatively low at the beginning, and it grows almost linearly until it reaches the level of about 97% after $15 - 20$ rounds.

Figures 7.5 and 7.6 present payoffs obtained by various types of banking agents against a *random static user*. The average payoffs at each round (from 1000000 independent games played for each setting) are presented. It is worth noting that the results are presented in a non-cumulative way, i.e. if the agent gains 1 in the first round of the game, and 3 in the second round, then his total gain in the first two rounds is 4.

The output of the two single-model agents (Figure 7.5) shows exactly why using many models of reality at the same time can be beneficial. $A(\textit{profile})$ – the agent who uses only the user's profile when making his decisions – loses quite a lot in the first few rounds, because the profile is often inaccurate after the first two or three responses from the user. Only after some 15 rounds the model of the user becomes accurate enough so that using it is more beneficial than playing the safest strategy all the time; and after $30 - 40$ rounds the agent approximates the real policy of the user quite correctly. $A(\textit{default})$, on the other hand, has a constant average payoff of 0.5 per round; since the user's policy is random, the expected value of playing the "safe offer" should be $\frac{1.5-1+1}{3} = 0.5$ indeed. The agent never loses much; however, he is not able to exploit the fact that the responses he gets from the user might tell something important about the user's real policy.

The hybrid agents play safer than $A(\textit{profile})$ with various degrees of success. The agent using fixed confidence of 0.9, i.e. $A(\textit{profile} + \textit{default}, C = 0.9)$, loses slightly less than $A(\textit{profile})$, but he gets suboptimal payoffs in the latter phase of the game, when other agents exploit their knowledge to a larger extent. Double-model agents using *Cbound*, *Csize* and *Cwang* play similar to $A(\textit{default})$ at the beginning of the game, and close to $A(\textit{profile})$ later on (which seems a natural consequence of the way the make their decisions). In general, the results show that an agent using such a hybrid model of the reality can be better off than an agent using either the profiles or the default user model alone.[1] *Cwang* (for $k = 1$) and *Cbound* (for $\mathfrak{m} = 2$) give best results, while *Csize* fares slightly worse despite quite complicated parameters setting ($\Lambda = 0.8$ and variable $\mathfrak{m} = 10 + N^{1.5}$). As the charts suggest – the agent does not lose money at the beginning of an interaction (because C is low and therefore he is using

[1] Unfortunately, it is not possible to present all the results on a single chart because the chart would be completely unreadable then.

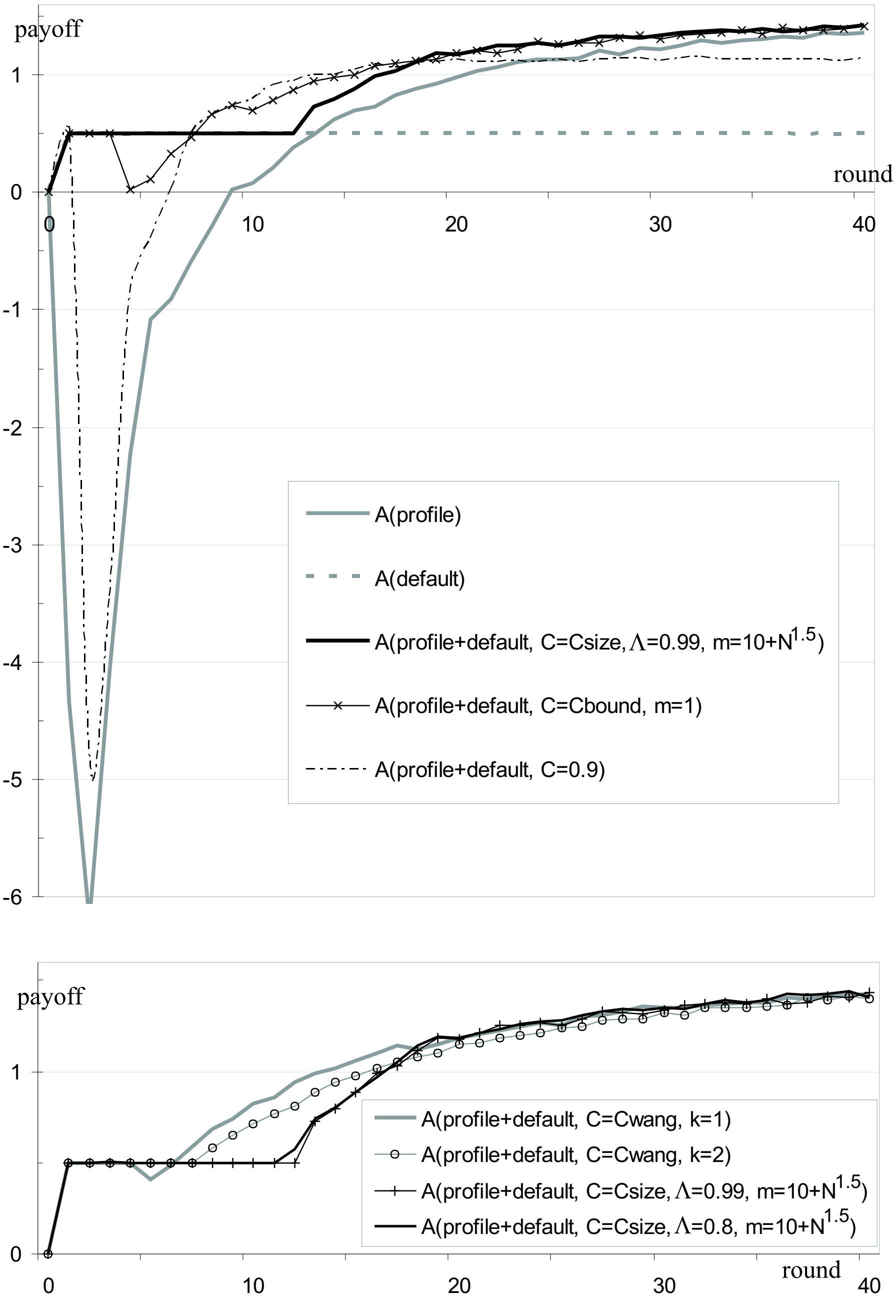


Figure 7.5: Hybrid agents vs. single-model agents: the average payoffs

mostly the default model). On the other hand, the confidence is almost 1 by the time the obtained knowledge becomes more accurate so the agent can start using the user profile successfully. Experiments with other payoff tables gave similar results (cf. Section 8). Moreover, it seems worth pointing out that, in the middle phase of the game, the hybrid agents earn more than *each* single-model agent. Thus, the average output of an agent using a linear combination of models is not necessarily a linear combination of the respective single-model agents' outputs. The hybrid agents seem to switch between safe and adaptive behavior accurately.

It should be obvious that the confidence needed to combine alternative models is neither meta-probability nor meta-likelihood. The simulations suggest that the practical uncertainty concerned here is rather related to the distance/deviation of the model from the reality in a way. Interestingly, an agent using $1 - expdev$ gets positively best payoff, especially in the initial phase of the interaction (see Figures 7.6 and 7.8). Perhaps the expected absolute deviation is not the best deviation measure for this purpose but it seems a close shot at least. Of course, we cannot *design* such an agent (because the agent can be hardly expected to know the $expdev$ value in the same way in which he has no direct access to the real policy of the user), but we can design an agent that tries to approximate the value, and use it afterwards.

The measures presented here are primarily designed to tackle lack of data, not the user's dynamics. However, some experiments with dynamic users have also been run. Simulating dynamic policies that imitate preferences drift of human users was not easy, because human agents are hardly completely random with respect to their policies. True, humans' preferences drift – and the drift is never completely predictable – but neither is it completely chaotic. Real users are usually committed to their preferences somehow, so the preferences drift more or less inertly (the drift changes its direction in a long rather than short run). Here, random initial and final policies p_0, p_{100} were generated for every simulation, and the user was changing his preferences from p_0 to p_{100} in a linear way: $p_i(b) = p_0(b) + \frac{i}{100}(p_{100}(b) - p_0(b))$. Figure 7.7 presents the confidence evolution for a dynamic user and $\lambda = 0.95$. Figure 7.8 shows the results of the "banking game" in the dynamic case. Here, the hybrid agent using *Cbound* fares best (except for the hypothetical agent using $C = 1 - expdev$), with other hybrid agents close behind. Most notably, the *Cbound* agent is never worse than both single-model agents. The agent using only the default model is omitted on the chart to make it clearer: as before, his average payoff has been about 0.5 per round all the time.[2] It should be noted that the learning method used in the experiments (i.e. counting with decay) is *not* linear, so it is not true that this particular type of user simulation dynamics was chosen to suit the learning algorithm.

7.3 Detecting Changes of Pattern: First Attempt

This section is focused on the second kind of an agent's uncertainty about his own beliefs: even provided a substantial amount of evidence or data, the agent may at some moment detect that the resulting model of the reality has ceased to characterize the

[2] A similar chart for the agent using $\lambda = 1$ shows the same regularities, although the payoffs are generally worse because the learning method is less flexible.

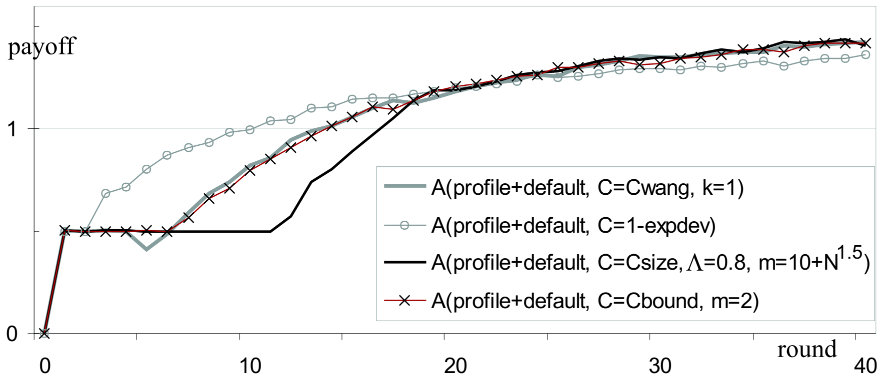


Figure 7.6: Hybrid agents vs. single-model agents: the average payoffs continued

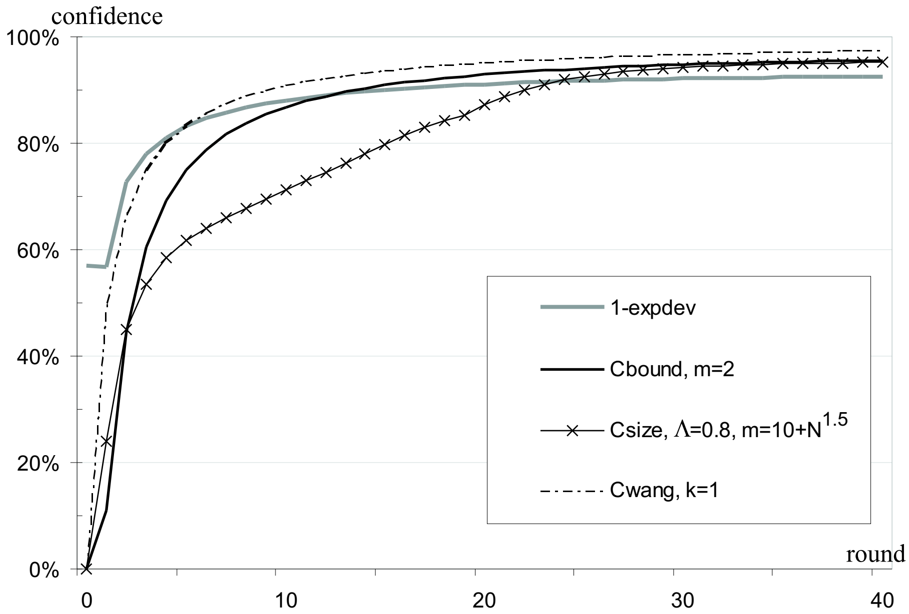


Figure 7.7: Confidence: interaction with a dynamic user, $\lambda = 0.95$

current behavior of the environment and/or other agents. In such case, one may suspect that the environment might have changed considerably, and the model is not consistent with its real source any more. The agent can certainly benefit from detecting conspicuous changes of pattern, and acting more cautiously in such situations.

Two confidence measures to capture this kind of meta-uncertainty are proposed and evaluated in this section and in Section 7.4. Some researchers from the probability

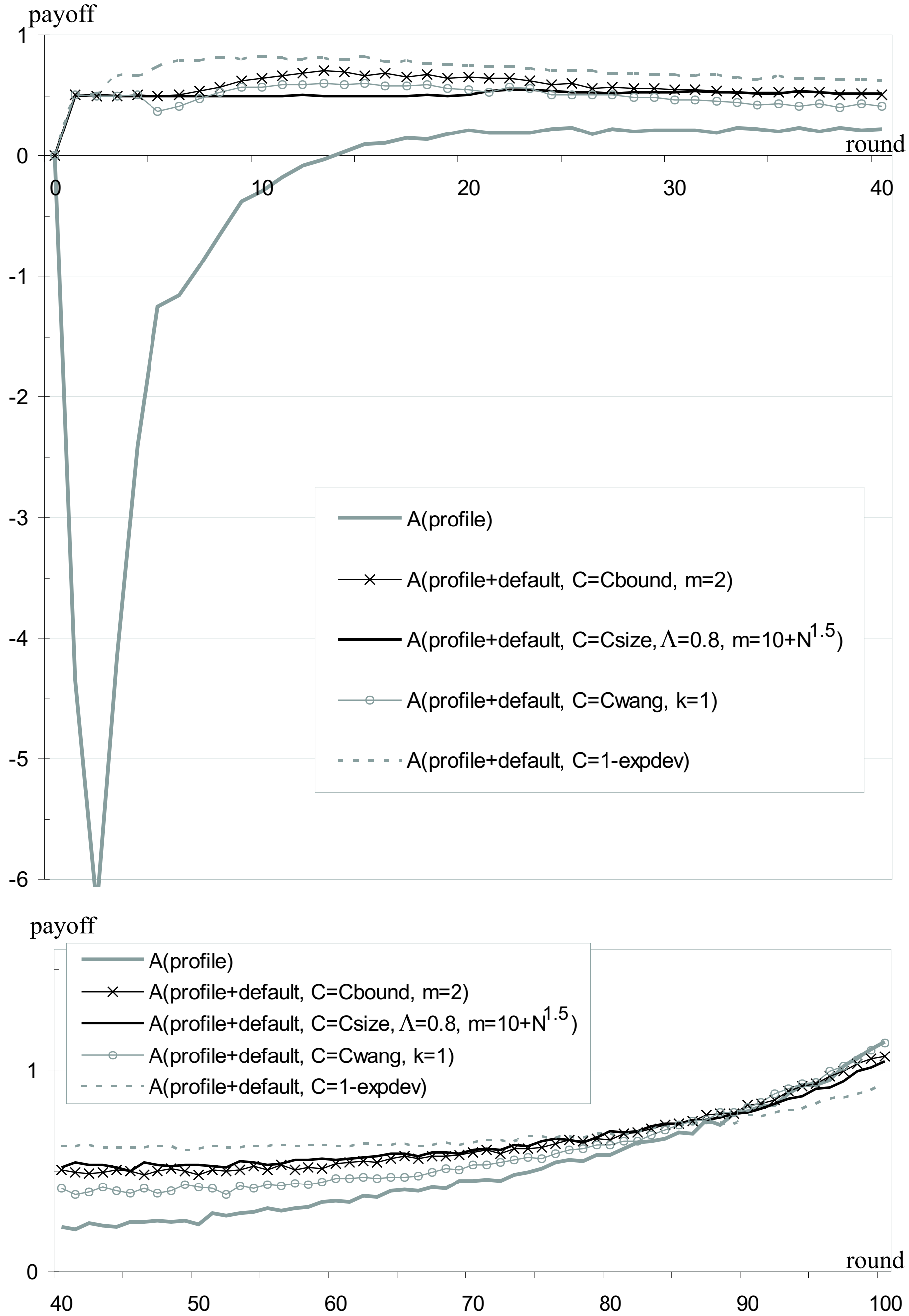


Figure 7.8: Hybrid agents vs. single-model agents: dynamic opponent, $\lambda = 0.95$

theory community suggest that – to solve the problem – we should take the agent's probabilistic knowledge as a random quantity, and use its variance as a clue (Pearl, 1987; Kyburg, 1988). Therefore the first measure being proposed is based on the aggregate variance of the model (Section 7.3.1). The second measure (Section 7.4) is based on the self-information loss function (or log-loss function), used widely in the fields of information theory and universal prediction (Merhav and Feder, 1998).

7.3.1 Aggregate Variance with Temporal Decay and the Variance-Based Confidence

To detect the moments when the user's behavior changes considerably, some dispersion measure can be used that captures its variability. The larger the dispersion, the smaller the confidence – thus, if the dispersion estimation is $disp \in [0,1]$, the confidence measure C can be somehow proportional to $1 - disp$. If we assume that the agent's knowledge $\hat{p}_i$ approximates the user's policy p_i at various time points $i = 1..n$, we can use the variability of the knowledge (over time) to estimate the variability of the real policy. It reflects also the following intuition: if the agent has collected enough data to estimate a policy of a static user, and still he has to update the estimation considerably, the pattern must be changing somehow.

The aggregate sample variance (for all the actions $b \in ActB$ the user can choose to proceed with)

$$\sum_{b \in ActB} V(\hat{p}_{i=1..n}(b)) = \frac{1}{n} \sum_b \sum_{i=1}^{n} (\hat{p}_i(b) - \overline{\hat{p}_{i=1..n}(b)})^2$$

has some welcome properties: it captures the variability of the *whole process* (the last update size does not), it can be computed incrementally (aggregate sample mean deviation can not), and its value is always between 0 and 1 (aggregate sample standard deviation is not). To implement a simple forgetting scheme, we use again the idea of the decay rate (cf. Section 7.2.2).

Definition 7.7 *Let* $C_V = (1 - disp_\Lambda)$*, where* $disp_\Lambda = \sum_b V_\Lambda(\hat{p}_{i=1..n}(b))$.

Proposition 7.11 $\frac{(1+\Lambda^n)(1-\Lambda)}{(1+\Lambda)(1-\Lambda^n)} \le C_V \le 1$.

Proof: The upper bound follows from the fact that $disp_\Lambda$ is a sum of nonnegative elements. The lower bound is a consequence of the following inequality:

$$\begin{aligned} disp_\Lambda &= \frac{1}{n}\sum_b \sum_{i=1}^{n} \hat{p}_i^2(b) - \frac{1}{n^2}\sum_b \big(\sum_{i=1}^{n} \hat{p}_i(b)\big)^2 = \\ &= 1 - \frac{1}{n} - (\frac{1}{n} - \frac{1}{n^2}) \sum_{i=1}^{n} \sum_{b_1 \ne b_2} \hat{p}_i(b_1)\hat{p}_i(b_2) - \frac{1}{n^2}\sum_b \sum_{i \ne j} \hat{p}_i(b)\hat{p}_j(b) \le \\ &\le 1 - \frac{(1+\Lambda^n)(1-\Lambda)}{(1+\Lambda)(1-\Lambda^n)} \end{aligned}$$

and the bound is tight for $n \le |ActB|$. □

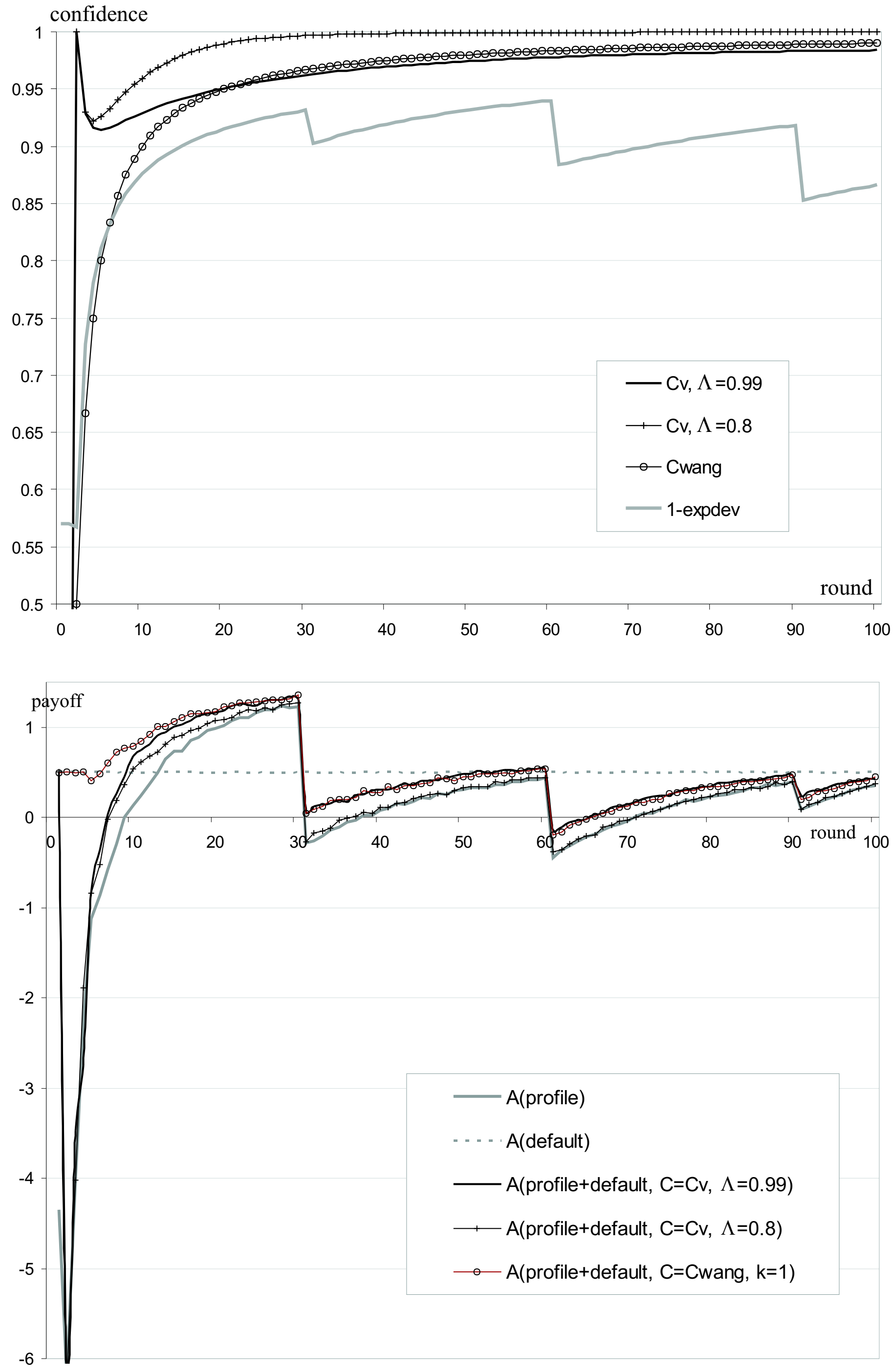


Figure 7.9: Variance-based confidence: (A) confidence values, (B) average payoffs against “stepping” users

7.3.2 Experimental Results for C_V

The aim of the simulations was to compare the output of the hybrid agent using C_V as the confidence to the hybrid agent using Wang's confidence $C_{Wang} = n/(n+1)$, and to the outputs of both single-model agents. 1000000 independent random interactions (a sequence of 100 rounds each) have been simulated for each agent. Figure 7.9A shows how the confidence values evolve for a dynamic user. The confidence values are compared against the expected absolute deviation of the learned profile from the real policy of the user: $expdev = \sum_b |\hat{p}(b) - p(b))| \cdot p(b)$, or rather the "accurateness" of the profile, i.e. $1 - expdev$. The dynamic opponents were simulated as "stepping" users: random initial and final policies p_0, p_{100} were generated for every simulation, and the user was changing his preferences every 30 steps: $p_i(b) = p_0(b) + (i \text{ div } 30)(p_{100}(b) - p_0(b))/3$.

Figure 7.9B shows the average payoffs, and suggests that an agent using such a hybrid model of the reality can be better off than an agent using either the profiles or the default user model alone. Again, such a "multi-model" agent does not lose so much money at the beginning of an interaction (because the confidence is low and therefore he is using mostly the default model); then the confidence increases almost to 1 and the agent's knowledge becomes more accurate at the same time – so the agent can start using the user profile successfully. The results of the simulations show that using the variance-based confidence improves the agent's payoff, but it is never better than C_{Wang}, which is all the more surprising because the Wang's measure was designed to tackle a *static* and not dynamic environment. The problem is that the "self-observation" approach proposed in the literature leans very heavily on the learning method being used by the agent. Usually the model gets stable quite fast regardless of the user's responses; any new item has a very small impact on the mean value after some, say, 20 observations have been collected. Therefore the measure does not really detect the changes in the user's behavior. It is easy to see in Figure 7.9A that C_V increases constantly almost all of the time, even when the model becomes less and less accurate, i.e. after the 30, 60 and 90th step. The agent may even focus on his most recent beliefs – it does not make things any better, because these are the beliefs that do not change.

Since the simulations show that the measure does not hold to its promise, an alternative measure is proposed in Section 7.4.

7.4 Detecting Changes via Logarithmic Loss Function

Log-loss function is used in the research on machine learning and time series prediction – especially universal prediction, where a prediction does not necessarily have to be a simple estimation of the next observation, but it can be a complex structure (a probability assessment, a strategy etc.), and the real underlying structure (the "source") generating the events is assumed to be unknown (Merhav and Feder, 1998). The universal prediction methods focus on finding a good predictor, not on assessing *how* good it is, though. A way of transforming the log-loss values into confidence values is proposed in this section.

7.4.1 Confidence Based on Self-Information Loss Function

Definition 7.8 *Let $\hat{p}_i$ represent the agent's beliefs about the preferences of the user in response to a particular action from the agent in a particular state of the environment (at the ith step of interaction within this context). Let b_i^* be the user's actual response at that step. One-step loss l_i and the average loss in n steps L_n can be defined using the log-loss function:*

$$\begin{aligned} l_i &= logloss(\hat{p}_i, b_i^*) = -\log_2 \hat{p}_i(b_i^*) \\ L_n &= \frac{1}{n}\sum_{i=1}^{n} l_i = -\frac{1}{n}\sum_{i=1}^{n} \log_2 \hat{p}_i(b_i^*) \end{aligned}$$

Note that the expected value of l is a function of two probability distributions: the real distribution p (the "source" distribution), and its model $\hat{p}$ built by the learning agent. More formally, $E\ l = -\sum_b p(b)\log_2 \hat{p}(b) = El(p, \hat{p})$. The loss is minimal (in the sense of expected value) when the agent has been guessing correctly, i.e. when the model he used was a true reflection of the reality: $\hat{p} = p$ (Merhav and Feder, 1998). However, this holds only if we assume that p is fixed, and not in general, as the following example demonstrates.

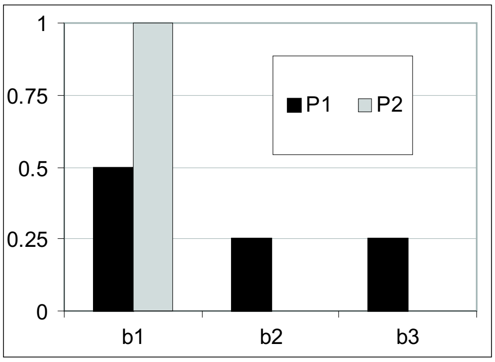


Figure 7.10: Minimal vs. optimal loss – an example

Example 7.1 Consider an agent who estimates the user's policy with probability distribution $\hat{p} = P1$ at some moment (see Figure 7.10). If his guess is right (i.e. the user's policy is $p = P1$ indeed), the expected loss is $E\ l = -0.5\log_2 0.5 - 2\cdot 0.25\log_2 0.25 = 1.5$. Yet if the real policy is $p = P2$, then $E\ l = -1\log_2 0.5 = 1$: the agent's loss can be smaller when his guess is wrong... In other words, $El(p, \hat{p})$ has a global minimum of $\hat{p} = p$ for a fixed p, but not when we consider all the possible source distributions. $\square$

Note that this is not a problem in time series prediction. The source distribution is presumably fixed in a single run (there is one *objective* source distribution), and hence $El(p, \hat{p})$ is *objectively* minimal for $\hat{p} = p$ (out of all the objectively possible values of $E\ l$). As long as the agent is not interested in the loss values themselves – only in finding the minimum point – minimizing the mean L_n is a valid strategy for him to find a model $\hat{p}$ that approximates the true probability distribution p. However, when the source distribution is unknown, some smaller loss values may be deemed possible from the agent's *subjective* point of view. Moreover, he may experience a smaller loss in a subsequent interaction in which his beliefs would be actually farther from the reality.

Example 7.2 Consider the learning agent from Example 7.1 again. Suppose the agent computes his *disconfidence* in his own model of the reality as a value proportional to L_n. Suppose that he interacts with two users, and in both cases he gets $\hat{p} = P1$. Moreover, let the real policy of the first user be $p = P1$, and the second: $p' = P2$. In a long run, our agent is going to obtain average loss of 1.5 in the first case, and 1 in the second. Thus, he is going to trust his beliefs more in the latter (where he actually guessed the policy incorrectly) – which is unacceptable. In consequence, the minimal loss is not the optimal loss in this case. □

What does "optimal" mean then? Let us define the *optimal series of models* as a sequence of the true probability distributions: $\hat{p}_i = p_i$ for $i = 1..n$.

Definition 7.9 *The* optimal expected loss Opt_n *is the expected value of the average loss we get provided the actual sequence of models* $\hat{p}_1..\hat{p}_n$ *is optimal.*

Definition 7.10 *The* loss deviation Δ_n *is the difference between the actual loss* L_n *and the optimal loss* Opt_n*.*

Note that:

$$\begin{aligned}
Opt_n &= EL_n = -\sum_{(b_1..b_n)} \left(p(b_1..b_n)\frac{1}{n}\sum_{i=1}^{n}\log_2 \hat{p}_i(b_i)\right) = \\
&= -\frac{1}{n}\sum_{i=1}^{n}\sum_{b_i} p_i(b_i)\ \log_2 \hat{p}_i(b_i) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{b} \hat{p}_i(b)\ \log_2 \hat{p}_i(b) \\
\Delta_n &= L_n - Opt_n = -\frac{1}{n}\sum_{i=1}^{n}\left(\log_2 \hat{p}_i(b_i^*) - \sum_{b}\hat{p}_i(b)\ \log_2 \hat{p}_i(b)\right)
\end{aligned}$$

Now, the loss deviation Δ_n (or rather its absolute value) seems a better basis for the confidence than the loss itself. As different $\hat{p}$'s give different loss characteristics, however, they also define very different deviation intervals. For $\hat{p}_i = P2, i = 1..n$, for instance, the only possible values for Δ_n are 0 and ∞ – if the model has proved to be even slightly mistaken, then Δ_n will remain ∞ forever. On the other end of the scale it is easy to observe that if the agent stubbornly keeps the uniform distribution as the user's model (i.e. $\hat{p}(b) = \frac{1}{|ActB|}$ all the time), then the deviation Δ_n is always 0, regardless of the actual responses from the user. In both cases the value of Δ_n tells virtually nothing about the actual reliability of $\hat{p}$. It would be desirable that our confidence measure produced more appropriate values, or at least "signal" such situations instead of giving unreliable output.

Between both extremes the range of possible Δ_n also vary: it is close to $(0, \infty)$ for very unbalanced models, and very narrow when $\hat{p}$ is close to the uniform distribution. It is proposed here that we can normalize the loss deviation with its range ($\Delta_n^{max} - \Delta_n^{min}$) to obtain disconfidence value that does not depend on the actual models $\hat{p}$ so much.

Definition 7.11 *The log-loss-based confidence measure can be defined as:*

$$\begin{aligned} C_{log} &= 2^{-\left|\frac{\Delta_n}{\Delta_n^{max}-\Delta_n^{min}}\right|} \qquad \textit{where} \\ \Delta_n^{max} &= \max_{(b_1^*..b_n^*)} \{\Delta_n\} \\ \Delta_n^{min} &= \min_{(b_1^*..b_n^*)} \{\Delta_n\}. \end{aligned}$$

Note that:

$$\begin{aligned} \Delta_n^{max} &= \max_{(b_1^*..b_n^*)} \{L_n - Opt_n\} = \max_{(b_1^*..b_n^*)} \{L_n\} - Opt_n \\ \Delta_n^{min} &= \min_{(b_1^*..b_n^*)} \{L_n - Opt_n\} = \min_{(b_1^*..b_n^*)} \{L_n\} - Opt_n \\ \Delta_n^{max} - \Delta_n^{min} &= \max_{(b_1..b_n)} \{-\frac{1}{n}\sum_{i=1}^{n} \log_2 \hat{p}_i(b_i)\} - \min_{(b_1..b_n)} \{-\frac{1}{n}\sum_{i=1}^{n} \log_2 \hat{p}_i(b_i)\} \\ &= -\frac{1}{n}\sum_{i=1}^{n} \min_{b_i} \log_2 \hat{p}_i(b_i) + \frac{1}{n}\sum_{i=1}^{n} \max_{b_i} \log_2 \hat{p}_i(b_i) \\ &= \frac{1}{n}\sum_{i=1}^{n} \left[\log_2 \max_b \hat{p}_i(b) - \log_2 \min_b \hat{p}_i(b)\right] \\ &= \frac{1}{n}\sum_{i=1}^{n} \log_2 \frac{\max_b \hat{p}_i(b)}{\min_b \hat{p}_i(b)} \end{aligned}$$

Proposition 7.12 *The measure has the following properties:*

1. *$n\ \Delta_n$ and $n\ (\Delta_n^{max} - \Delta_n^{min})$ can be computed incrementally – the agent does not have to keep any additional information;*

2. *if the value of C_{log} can be computed, then $0.5 \leq C_{log} \leq 1$;*

3. *C_{log} is undefined exactly in the two cases where Δ_n is most dubious: when $\hat{p}_i$'s are uniform for all $i = 1..n$ or when there exist i and b such that $\hat{p}_i(b) = 0$. Note also that, when $\hat{p}_i$ are frequency distributions (or, more generally: probability distributions obtained through Bayesian updating), the first situation can happen only at the very beginning of the interaction, i.e. for $i = 1$. Moreover, the agent can be prevented from the latter situation by starting from an initial distribution such that $\hat{p}_1(b) > 0$ for every b (for instance, he may use the uniform rather than nil distribution as the starting point). Then we make sure that the probabilities will always be positive.*

7.4.2 Log-loss Confidence with Temporal Decay

Temporal decay can be introduced into the log-loss confidence to make the recent loss values matter more than the old ones: we can redefine the average loss L_n to be a mean with decay, and again base the confidence on the relative loss deviation.

Definition 7.12 *Let us define the average long-term loss with decay as* $L_n^\Lambda = M_\Lambda(l_{i=1..n})$*, the optimal expected loss with decay as* $Opt_n^\Lambda = EL_n^\Lambda\,[p \leftarrow \hat{p}]$*, and the decayed loss deviation as* $\Delta_n^\Lambda = L_n^\Lambda - Opt_n^\Lambda$*. Moreover, let* $\Delta_n^{max,\Lambda} = \max_{(b_1^*..b_n^*)}\{\Delta_n^\Lambda\}$ *and* $\Delta_n^{min,\Lambda} = \min_{(b_1^*..b_n^*)}\{\Delta_n^\Lambda\}$*. Now:*

$$C_{log}^\Lambda = 2^{-\left|\frac{\Delta_n^\Lambda}{\Delta_n^{max,\Lambda} - \Delta_n^{min,\Lambda}}\right|}$$

Again,

$$\begin{aligned}\Delta_n^\Lambda &= L_n^\Lambda - Opt_n^\Lambda = \frac{\sum_{i=1}^n \Lambda^{n-i}[-\log_2 \hat{p}_i(b_i^*) + \sum_b \hat{p}_i(b)\log_2 \hat{p}_i(b)]}{\sum_{i=1}^n \Lambda^{n-i}} = \\ &= -M_\Lambda\big(\log_2 \hat{p}_i(b_i^*)\big)_{i=1..n} + M_\Lambda\big(\sum_b \hat{p}_i(b)\log_2 \hat{p}_i(b)\big)_{i=1..n}\end{aligned}$$

and

$$\begin{aligned}\Delta_n^{max,\Lambda} - \Delta_n^{min,\Lambda} &= \max_{(b_1^*..b_n^*)}\{-M_\Lambda(\log_2 \hat{p}_i(b_i^*))\} - \min_{(b_1^*..b_n^*)}\{-M_\Lambda(\log_2 \hat{p}_i(b_i^*))\} = \\ &= \frac{-\sum_{i=1}^n \Lambda^{n-i}\log_2 \min_{b_i^*}\hat{p}_i(b_i^*)}{\sum_{i=1}^n \Lambda^{n-i}} + \frac{\sum_{i=1}^n \Lambda^{n-i}\log_2 \max_{b_i^*}\hat{p}_i(b_i^*)}{\sum_{i=1}^n \Lambda^{n-i}} = \\ &= \frac{\sum_{i=1}^n \Lambda^{n-i}\log_2[\max_b \hat{p}_i(b)/\min_b \hat{p}_i(b)]}{\sum_{i=1}^n \Lambda^{n-i}} = M_\Lambda\big(\log_2 \frac{\max_b \hat{p}_i(b)}{\min_b \hat{p}_i(b)}\big)\end{aligned}$$

Remark 7.13 C_{log}^Λ *retains the properties of* C_{log} *(cf. Proposition 7.12).*

7.4.3 Experiments

We ran a number of simulations, analogous to the ones in Section 7.3.2, in order to verify the new confidence measure. Figure 7.11 shows how the confidence values evolve against a "stepping" user. The confidence values are compared against the expected absolute deviation of the learned profile from the real policy of the user: $expdev = \sum_b |\hat{p}(b) - p(b))| \cdot p(b)$, or rather the "accurateness" of the profile, i.e. $1 - expdev$. It can be observed that the logloss-based measure is able to detect changes in the user's behavior – at least when temporal decay is employed. Every time the user changes his preferences (that is, after each 30 rounds), the confidence value $C_{log}^{0.9}$ decreases and starts to grow only when the model becomes closer to the reality again. Thus, we finally come up with a measure that has some substantial advantage over the Wang's measure, which increases always in the same manner regardless of the users' responses (because it was designed for static sources of data).

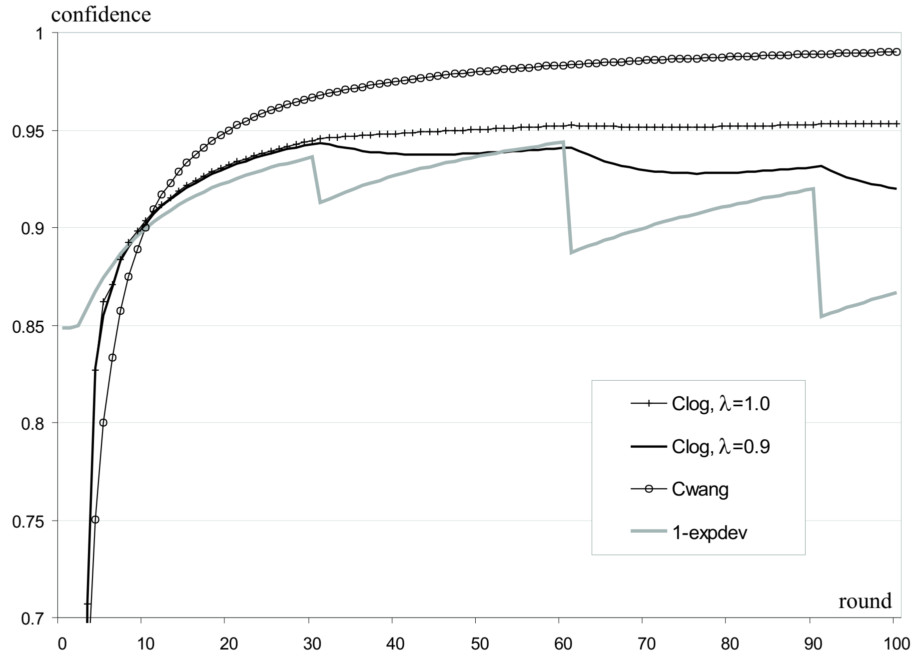


Figure 7.11: Confidence values: C_{log} vs. C_{Wang} vs. accurateness

Figure 7.12 shows – again – that an agent using a hybrid model of the reality does not lose so much when his knowledge is inaccurate (i.e. at the beginning of the game, and after each change of the user's policy), the agent starts using it successfully. Moreover, the agent still has potential to adapt to the changes of the environment. In most cases there were no significant differences in the output of the agents using C_{log} with or without temporal decay.

7.4.4 A Slightly Different Log-loss Confidence

The loss values that we "accumulate" to compute C_{log} are based on the logarithmic function due to its useful properties. However, we need the confidence values to span the set of $[0, 1]$, which calls for a "translation" of the relative loss deviation values back to the interval. Defining the log-loss confidence in Section 7.4.1, we used the exponential function 2^x to achieve the goal. Of course, there are many different translations that accomplish this, and the alternative presented below is probably the simplest among them.

Definition 7.13 *Let us define the alternative log-loss confidence as:*

$$C_{log}^{*\Lambda} = 1 - \left| \frac{\Delta_n^{\Lambda}}{\Delta_n^{max,\Lambda} - \Delta_n^{min,\Lambda}} \right|$$

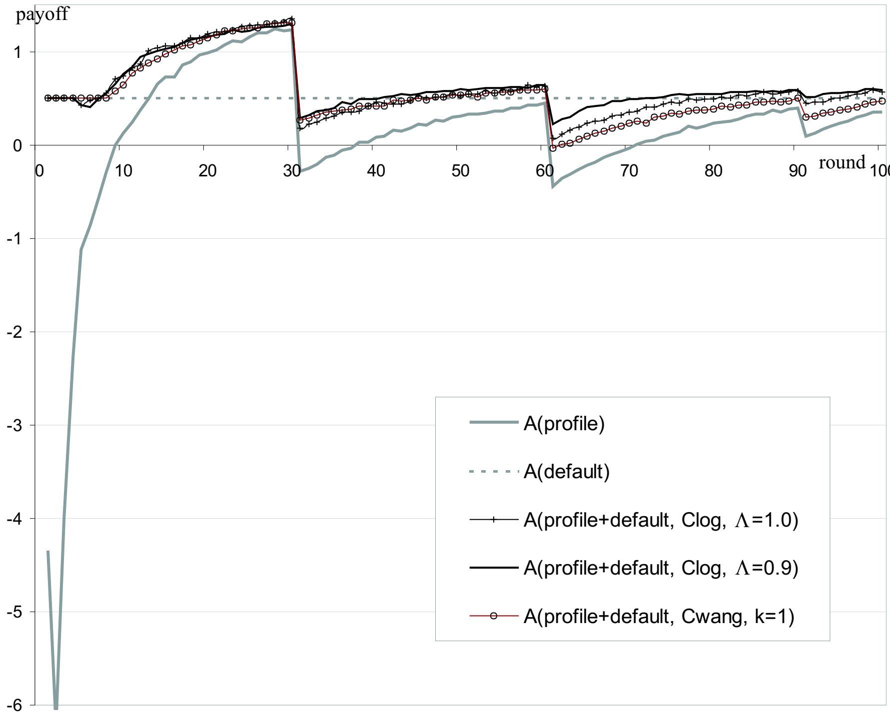


Figure 7.12: Hybrid agents vs. single-model agents: payoffs against "stepping" users

$C^{*\Lambda}_{log}$ has similar properties as C^{Λ}_{log}, but it spans the whole interval of $[0, 1]$. The way the confidence values evolve is shown in Figure 7.13. We note that changes of C^{*}_{log} indicate changes of the profile accurateness even clearer than in the case of C_{log}. More results of experiments with hybrid agents using C^{*}_{log} can be found in Chapter 8.

Proposition 7.14 *The measure has the following properties:*

1. *$n\,\Delta^{\Lambda}_{n}$ and $n\,(\Delta^{max,\Lambda}_{n} - \Delta^{min,\Lambda}_{n})$ can be computed incrementally;*
2. *if the value of $C^{*\Lambda}_{log}$ can be computed, then $0 \leq C^{*\Lambda}_{log} \leq 1$;*
3. *$C^{*\Lambda}_{log}$ is undefined only when $\hat{p}_i$'s are uniform for all $i = 1..n$ or when there exist i and b such that $\hat{p}_i(b) = 0$. Thus, if $\hat{p}_i$ are obtained through Bayesian updating and $\hat{p}_1(b) > 0$ for every b, we have that $C^{*\Lambda}_{log}$ can be computed for every $i \geq 2$.*

7.5 Confident Remarks

The experiments showed that a confidence measure can be useful – at least in some settings – for instance, to detect changes in a user's behavior, or as a means for weight-

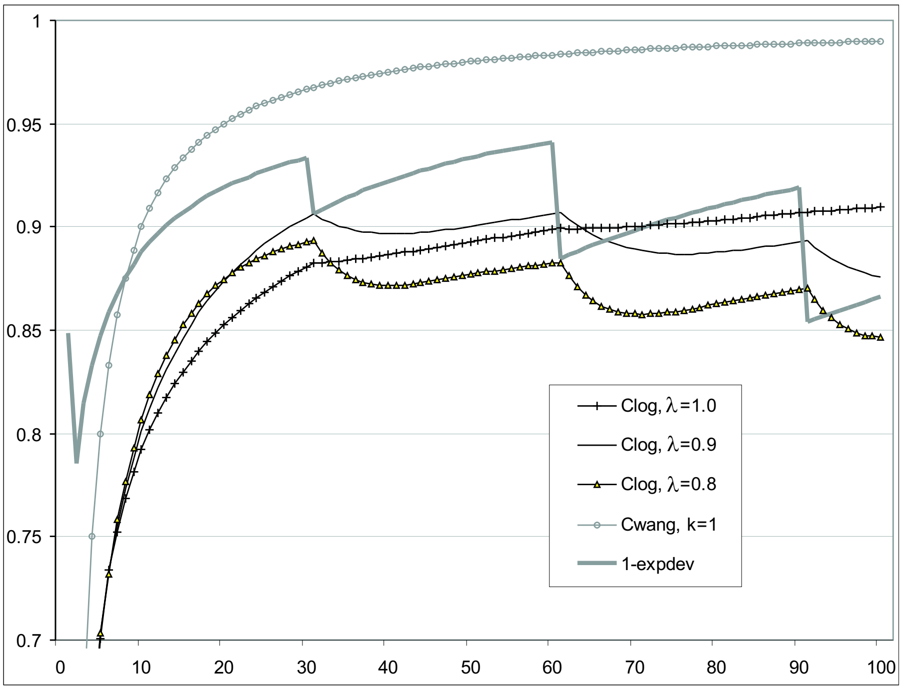


Figure 7.13: Confidence values: C^*_{log} vs. C_{Wang} vs. accurateness

ing alternative beliefs. Two kinds of measures have been proposed: *Cbound*, *Csize* and C_V represented the "self-observation" approach and were based on the variance of model evolution, while C_{log} was based on the logarithmic loss function. Some of the measures (namely, *Cbound* and *Csize*) were meant to capture the meta-uncertainty an agent should have when the amount of available data is insufficient. The others (C_V and C_{log}) addressed the problem of detecting conspicuous changes of patterns in the responses the agent receives from the world outside.

The simulations showed some merit behind *Cbound*, especially with the "magnifying glass" parameter set to $\mathfrak{m} = 2$. The experiments with *Csize*, on the other hand, revealed important deficiencies: even a quite complicated "tuning" scheme ($\mathfrak{m} = 10 + N^{1.5}$) plus an additional parameter Λ did not guarantee a substantial gain. Moreover, neither *Csize* nor *Cbound* performed much better than an extremely simple confidence measure *Cwang* $= N/(N + 1)$, proposed by Wang. The "self-observation" approach is probably too solipsistic – and bad learners yield incorrect self-assessment in consequence. C_V seems to suffer form the same condition.

Fortunately, a preliminary investigation of C_{log} brought encouraging results. The logarithmic loss function provides a direct link between the model and the new observations, and the temporal decay scheme lets the agent focus more on the results of recent predictions rather than all of them. In consequence, the measure is flexible enough to react appropriately even after many steps of collecting and analyzing data.

The results of the simulations suggest also that a meta-uncertainty measure, aimed for an agent who uses a hierarchy of beliefs, can be somehow based on the estimated deviation of the model from the reality – and definitely not on the meta-probability of the model correctness, or even the (potential) model variability over time.

Chapter 8

Safer Decisions against a Dynamic Opponent

SYNOPSIS. *Throughout the thesis, a considerable amount of knowledge has been put forward. Various game-like logics and models for multi-agent situations were found closely related; others yielded counterintuitive properties which made us try to revise them. In an attempt to combine safe game theory-based optimality criteria with adaptive machine learning solutions, the idea of multi-model decision making was proposed. Several confidence measures were investigated for an agent who may need to evaluate the accurateness of his own knowledge. Thus, the agent can assess his view of the environment, and combine safe play with exploiting the knowledge gathered so far.*

We resist the temptation to apply the same technique to the picture of multi-agent systems that emerged within this thesis, and assess its accurateness with one of the confidence measures. Instead, we opt for the safe strategy of presenting some more experiments with the simple e-banking agent. Finally, for those who are already fed up with the agent and his not-too-clever customers, some examples are shown of how ATL *models and planning can be used within the hierarchy of models.*

8.1 Introduction

The idea of hierarchical modeling of the reality was proposed in Chapter 6; Chapter 7 complemented it with research on a suitable confidence measure that can provide weights for the decision-making process. The aim of this chapter is to investigate the performance of such "multi-model" agents in the simplest possible case, i.e. in the case of an agent using exactly two alternative models of reality. The output of such a hybrid agent can be then compared with the performance of both "single-model" agents alone to see if (and when) a software agent can really benefit from using a more complicated belief structure and decision making scheme.

The controversy between normative models (like non-cooperative equilibria from game theory) and adaptive models (obtained through some kind of learning) has been an important inspiration for this part of the thesis. The adaptive solutions are more useful when the domain is cooperative or neutral; they also allow the agent to exploit deficiencies of his adversaries. The "best defense" assumptions are still attractive, though, in a situation when the agent risks real money. Even one enemy who plays his optimal strategy persistently can be dangerous then. Chapters 6 and 7 – and this chapter – present an attempt to integrate both approaches. The main model used by the agent in the experiments is a profile of the user; the other model is based on the maxmin equilibrium.

Section 8.2 extends the experiments presented in the two previous chapters, and presents them in a more systematic way. The results suggest that a software agent can combine machine learning with game theory solutions to display more profitable (or at least safer) performance in many cases. The confidence measure used here is not perfect, as the results of the simulations show. Further experiments should include also agents using more sophisticated learning methods.

This chapter builds on (Jamroga, 2003c) and uses some material from (Jamroga, 2003a).

8.2 The E-Banking Game

In this section, we present more experimental results that extend and complement the simulations already described in the two previous chapters. But first, let us recall and summarize the idea behind the "e-banking game", and the rules of the game.

8.2.1 Online Banking Scenario

The experiments reported in this section follow the scenario presented in Section 6.3.2: a software agent is designed to interact with users on behalf of an Internet banking service; he can make an offer to a user, and the user's response determines his payoff at this step of interaction.

The Game

The agent has 3 possible offers at hand: the "risky offer", the "normal offer" and the "safe offer", and the customer could respond with: "accept honestly", "cheat" or "skip". The complete table of payoffs for the game is given below. The table, presented in Figure 8.1, differs slightly from the one used in Chapters 6 and 7, yet the payoffs obtained by the agents are similar. Other payoff tables are tried in Sections 8.2.2 and 8.2.4, again with analogous results.

	accept	cheat	skip
risky offer	30	-100	0.5
normal offer	10	-30	0.5
safe offer	0.5	0	0.5

Figure 8.1: Payoffs for the game

The Agent

The banking agent is an adaptive 1-level agent, hence he maintains a model of the other agent's preferences, and uses the model in the decision-making process.[1] The agent estimates the user's policy p with a probability distribution $\hat{p}$, computed through simple Bayesian updating (Mitchell, 1997) with no observational decay:

$$\begin{aligned} \hat{p}_{i+1}(b) &= \begin{cases} \frac{\hat{p}_i(b)n_i+1}{n_i+1} & \text{if } b = b^* \\ \frac{\hat{p}_i(b)n_i}{n_i+1} & \text{if } b \neq b^* \end{cases} \\ n_{i+1} &= n_i + 1 \end{aligned}$$

where b^* is the actual response from the user in the last (ith) round of interaction. Value $n_0 \geq 0$ is the number of "virtual" training examples. The initial distribution $\hat{p}_0$ is uniform in most experiments, although the "safe" distribution (corresponding to the maxmin strategy) has also been tried.

At the same time the agent computes a confidence value C for the profile acquired so far. The default model is defined in the game theory fashion: the user is assumed an enemy who always cheats. The (multi-model) evaluation of every action a is based on sub-evaluations derived from both models separately:

$$eval(a) = C\ eval_{\mathit{profile}}(a) + (1 - C)\ eval_{\mathit{default}}(a).$$

The agent chooses action a with maximal $eval(a)$ (Section 8.2.2) or uses a more sophisticated decision making scheme to tackle the exploration-exploitation tradeoff (Section 8.2.3). Thus, the banking agent's hierarchy of beliefs consists of two models of the reality, presented in Figure 8.2.

Confidence

In order to provide the agent with a way of computing his "self-confidence" C, two measures are combined: the log-loss based confidence C^{Λ}_{log} (Section 7.4.1),[2] and the datasize-related measure proposed by Wang (Wang, 2001).

C^{Λ}_{log} helps to detect changes in the user's policy, but it is unreliable when the number observations is small. This disadvantage can be tackled, though, with the simplest

[1] Recall: a 1-level agent is an agent that models other agents as 0-level agents, i.e. agents whose behavior can be described with a probabilistic policy (Vidal and Durfee, 1998).

[2] We tested also the performance of an agent using the other version of the log-loss confidence – $C^{*\ \Lambda}_{log}$ from Section 7.4.4. Every time the results refer to the agent using C^*_{log} instead of C_{log}, we state it explicitly in the text.

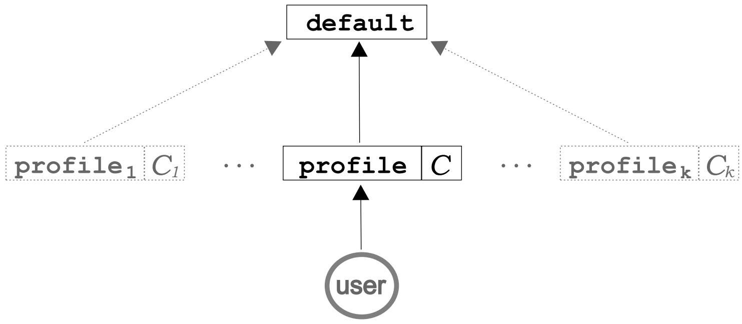


Figure 8.2: The simplest hierarchy: two models of reality at a time (repeated)

variant of the measure by Wang (for $k = 1$): $C_{Wang} = n/(n + 1)$. Now, the agent is confident in his knowledge if he has enough data *and* he detects no irregularities in the user's behavior:

$$\mathbf{C} = \min(\mathbf{C}_{\mathbf{log}}^{\mathbf{\Lambda}}, \mathbf{C}_{\mathbf{Wang}}).$$

The knowledge decay rate Λ was set to 0.9 throughout the experiments.

Users

To investigate performance of the hybrid agent, several series of experiments were run. The e-banking agent played with various kinds of simulated "users", i.e. processes displaying different dynamics and randomness. Those included:

- static (or stationary) 0-level user with a random policy: $p_0 = p_1 = \ldots = p_{100}$ generated at random at the beginning of each interaction;
- "linear" user: a dynamic 0-level agent with the initial and the final preferences p_0, p_{100} generated at random, and the rest evolving in a linear way: $p_i = p_0 + (p_{100} - p_0)/100$;
- "stepping" user: same as the "linear" one except that the preferences change after every 30 steps: $p_i(b) = p_0(b) + (i \text{ div } 30)(p_{100}(b) - p_0(b))/3$;
- "cheater": a user that chooses action "cheat" with probability of 1.0,
- "malicious": an adversary 0-level user with a stationary random policy for the first 30 rounds, then switching to the "cheater" policy.

The user types we consider most important are: the static user, the "stepping" user, and the "malicious" user. The static user defines the limit case with no dynamics at all, hence he is a perfect subject for machine learning algorithms. While our method of multi-level decision making is mostly aimed at inducing safer play in the context of *dynamic* users and environments, it is still interesting to check how it performs in this borderline case. The "stepping" user is aimed to mimic the interest drift of an ordinary indifferent human user. We believe that people are to some extent persistent with their preferences: the do not change them too often, and the direction of the changes is not completely random. Thus, the policy of the user changes only from time to time, and the changes follow in the same direction during a particular interaction. Finally, the "malicious" user represents an adversary entity that tries to squeeze from us as much as he can; every e-banking or e-commerce service must take such a possibility into account. The "malicious" user is more clever than the rest: he lets our e-banking agent to build up a false model of his strategy, and only then he unveils his real plan. In fact, this kind of user is a 0-level agent only in the strict, technical sense of the term. Since his strategy is generally quite well suited against 1-level adaptive agents, he might be arguably classified as a 2-level agent as well.

1000000 independent random interactions (a sequence of 100 rounds each) have been simulated for every particular setting; the average results are presented in Sections 8.2.2 and 8.2.3.

8.2.2 Playing Against Single-Minded Users

The user has been assumed rather simple-minded in the first series of experiments, in order to get rid of the exploration-exploitation tradeoff.

Remark 8.1 *The* exploration-exploitation tradeoff *is a well-known phenomenon within the Machine Learning research. Basically, the main objective of a learning agent is to* exploit *his knowledge in order to obtain more profit. Thus, it seems perfectly reasonable for the agent to execute the decision with highest expected payoff. However, the agent risks in such a case that he might get stuck in a local optimum if his knowledge is not completely accurate. Moreover, choosing a particular action affects also the (future) quality of the agent's knowledge, and hence (indirectly) persistent executing of the highest expectancy choice may lead to suboptimal results in the future. Thus, the agent can be better off trying to* explore *the environment and keep his knowledge up-to-date.*

Consider, for example, our e-banking agent. Suppose that the "safe offer" poses highest expected payoff according to the initial model of the environment. If the agent only exploits *the model, he will never execute any other offer than the safe one – yielding no successful negotiations even if the actual user is honest and willing to cooperate. Obviously, some exploration is necessary to determine the kind of customer we are dealing with. On the other hand, exploratory actions can be very costly if financial risk is involved.*

Thus, it has been assumed that the user's response does not depend on the actual offer being made: $p(cheat)$, $p(accept)$ and $p(skip)$ are the same regardless of the offer (if he is dishonest, he cheats for a small reward as well as a big one, for instance). In consequence, no specific exploration strategy is necessary – every action the agent can choose will reveal exactly the same about the user's policy – so the agent can just maximize $eval(a)$ when making his decisions. Results for various types of users are presented in Figures 8.3 and 8.4. The hybrid agent is almost never worse than the agent using only the user's profile (even for the static user), and in the most risky moments he plays much safer than the latter. Most notably, the hybrid agent do not lose much

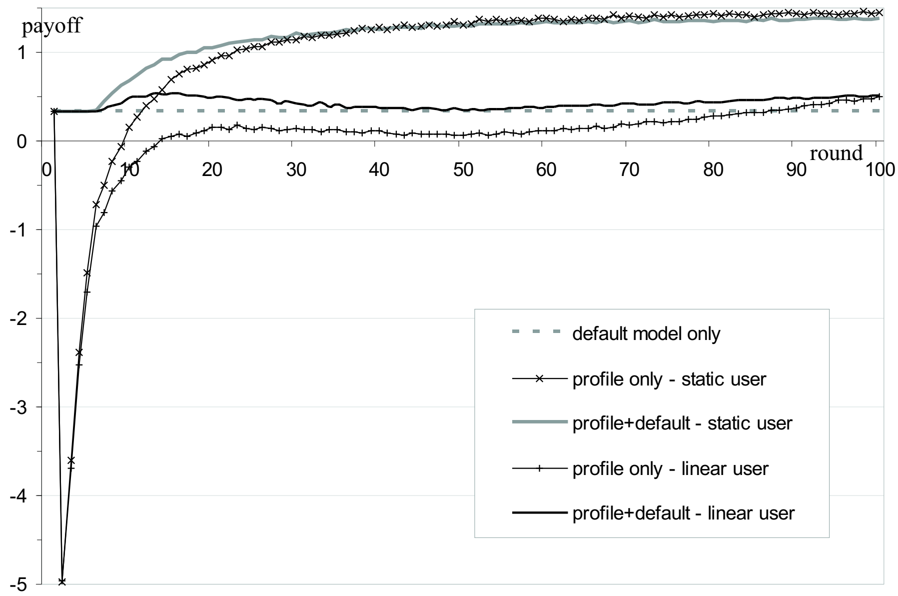


Figure 8.3: Hybrid agents vs. single-model agents: the average payoffs against single-minded users

score at the beginning of an interaction and in the moments when the user changes his policy (which is especially evident against the "malicious" user – cf. Figure 8.4). At the same time, the hybrid agent has potential to play positively better than the "default model" agent.

Some simulations were also run for a modified version of the banking game, representing a situation in which the agent's decisions involve less risk – see Figure 8.5. We tested also the hybrid agent using the alternative version of the log-loss confidence measure ($C^{*\Lambda}_{log}$ instead of C^{Λ}_{log}) – Figure 8.6 shows the outcome. Both series of experiments produced results similar to the ones before.

The "single-mindedness" assumption looks like a rough simplification. On the other hand, the preferences of a particular user (with respect to different offers) are hardly uncorrelated in the real world. For most human agents the situation seems to be somewhere between both extremes: if the user tends to cheat, he may cheat in many cases (although not all by any means); if the user is generally honest, he will rather not cheat (although the temptation can be too strong if the reward for cheating is very high). Therefore the assumption that the user has the same policy for all the agent's offers may be also seen as the simplest way of collaborative modeling (Zukerman and Albrecht, 2001). Section 8.2.4 gives some more rationale for this kind of assumption, while in the next section users with multi-dimensional (uncorrelated) policies are studied to complete the picture.

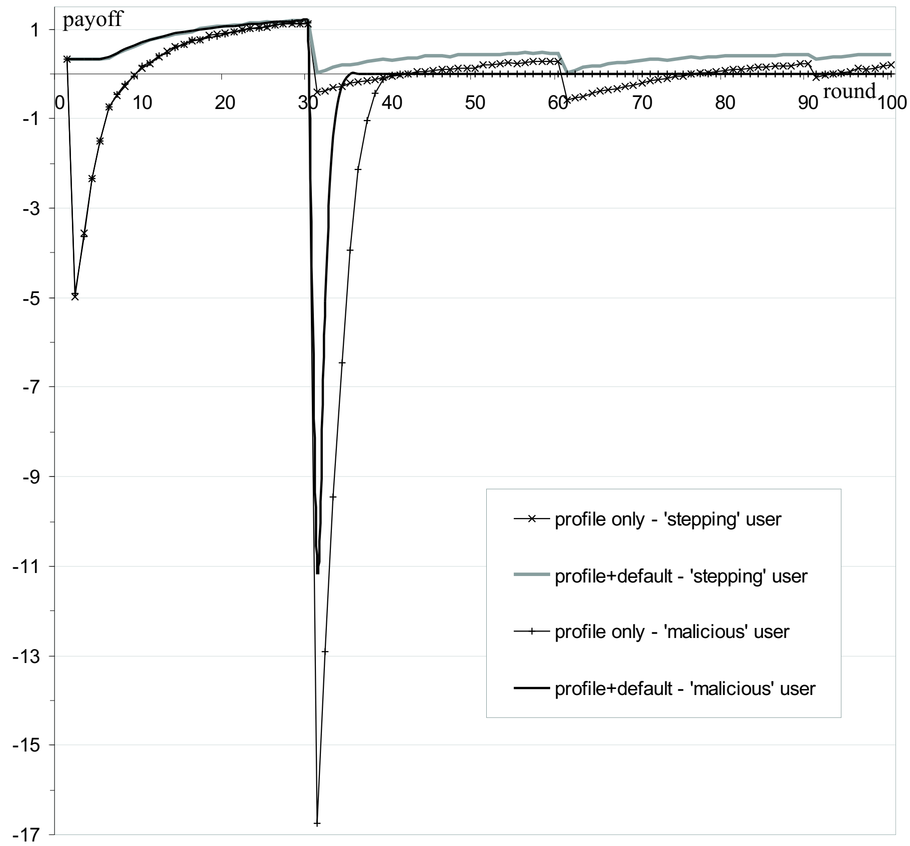


Figure 8.4: Hybrid agents vs. single-model agents: the average payoffs against single-minded users continued

8.2.3 Experiments for Users with More Complex Policies

Here, users were simulated with no restriction on the relation between their conditional policies $p(\cdot|\text{safe})$, $p(\cdot|\text{normal})$ and $p(\cdot|\text{risky})$. Boltzmann exploration strategy is one way to deal with the exploration-exploitation problem (Banerjee et al., 2000). The scheme uses the metaphor of the internal dynamics of a gas: as long as the temperature is high, the dynamics is also high, and the molecules display mainly random behavior; as soon as the gas starts to cool down, its behavior becomes more schematic and predictable. Putting our learning agent in a similar position, we usually assign him a "temperature" value that controls the randomness of his decision-making processes. The initial temperature T_0 is usually high to induce more random decisions, and thus more exploration of the environment at the beginning of the learning process. After that, the temperature is decreased by introducing a decay factor η, and updating the temperature with $T_n = \eta\, T_{n-1} = T_0\, \eta^n$.

	accept	cheat	skip
risky offer	30	-30	0.5
normal offer	10	-9	0.5
safe offer	0.5	0	0.5

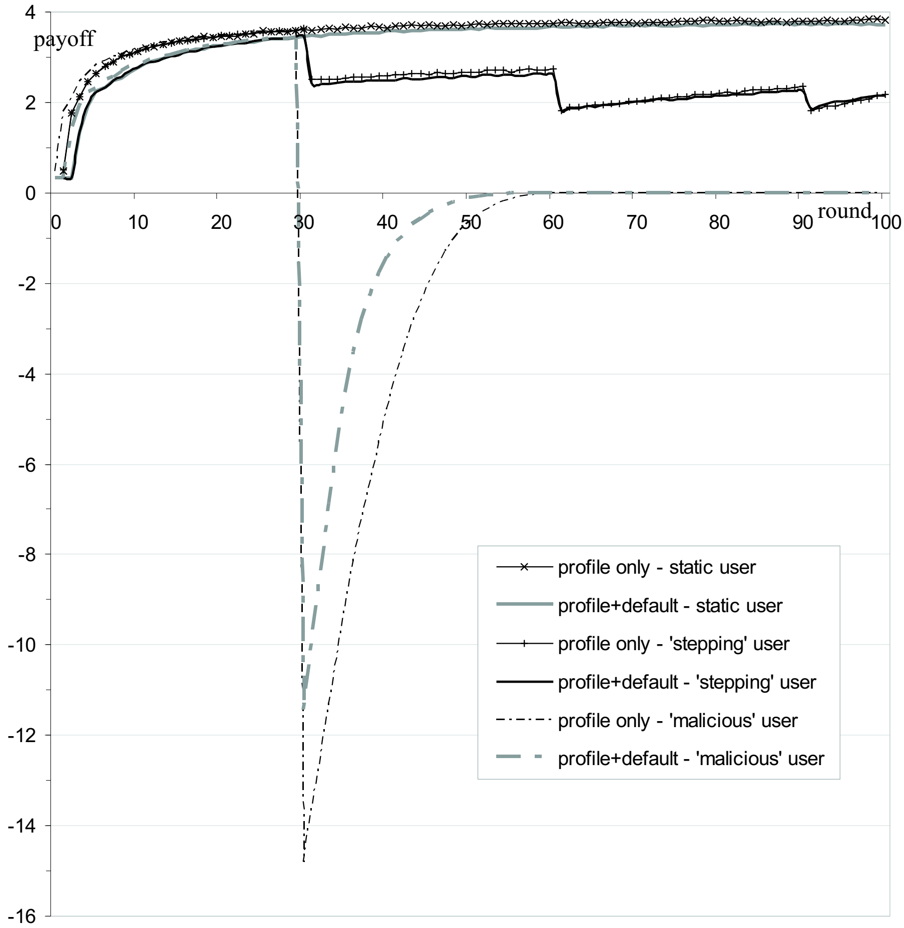


Figure 8.5: Payoff table and the results for the modified game

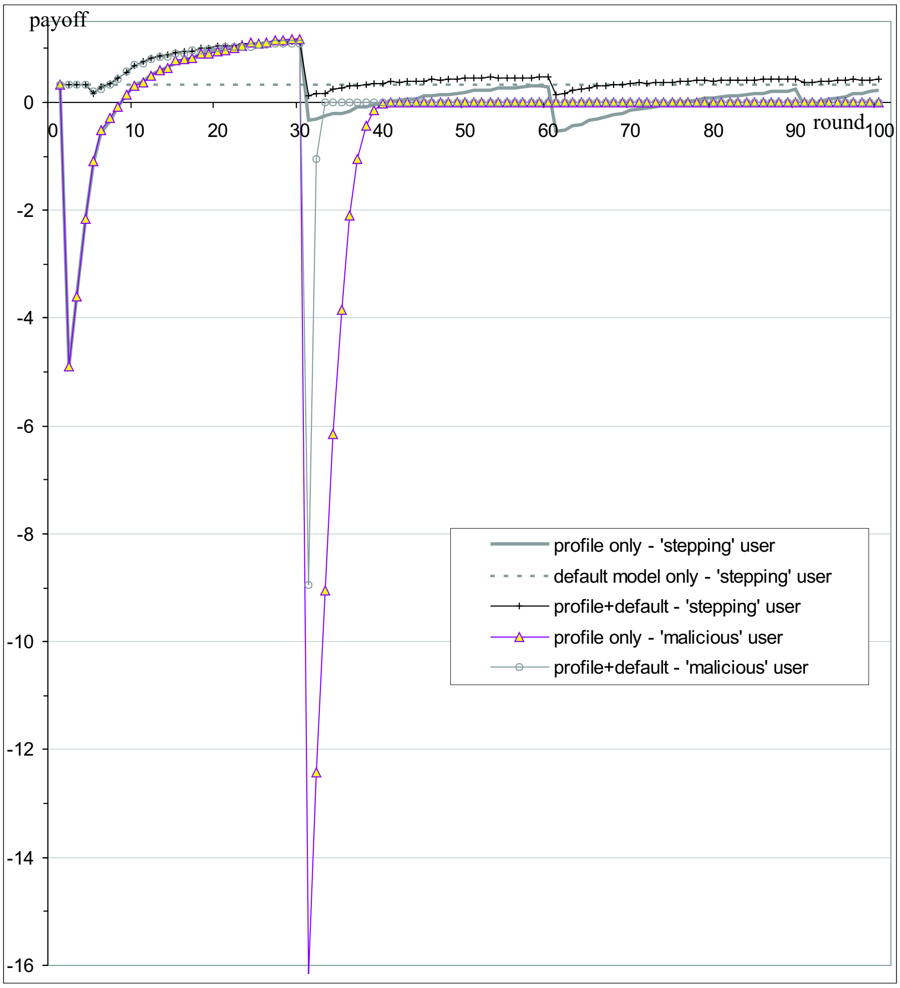


Figure 8.6: Hybrid agents vs. single-model agents: $C^{*\,0.9}_{log}$

Definition 8.1 (Banerjee et al., 2000) *Let T be the current value of the temperature parameter, and $eval(a)$ the current numerical evaluation of action a (e.g. its expected utility or, like in our case, its multi-model evaluation). An agent, using the Boltzmann exploration strategy, chooses action a semi-randomly with probability:*

$$P(a) = \frac{e^{eval(a)/T}}{\sum_{a'} e^{eval(a')/T}}$$

In other words, the agent employs a mixed strategy in which the actual probabilities are defined by the formula above.

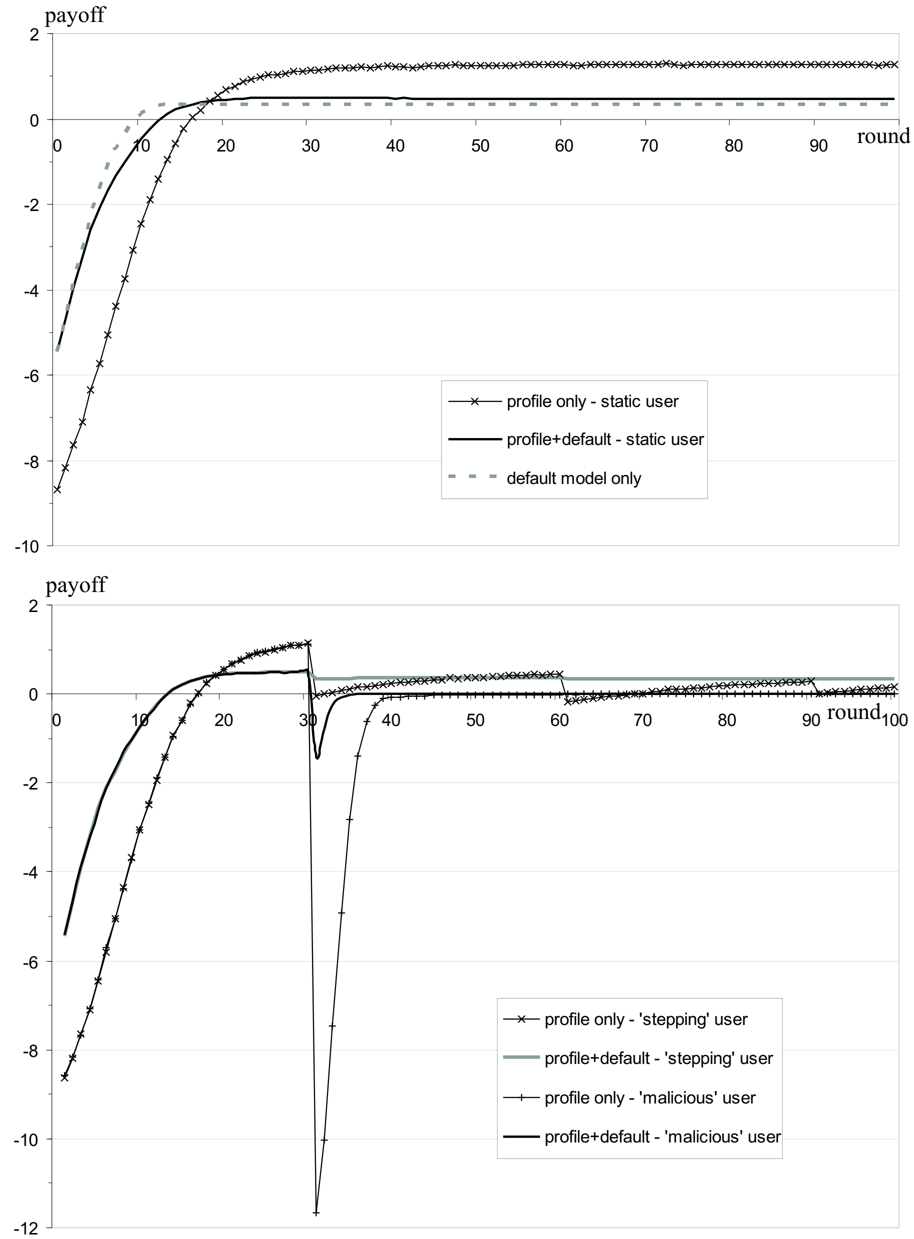


Figure 8.7: Playing against non-singleminded users

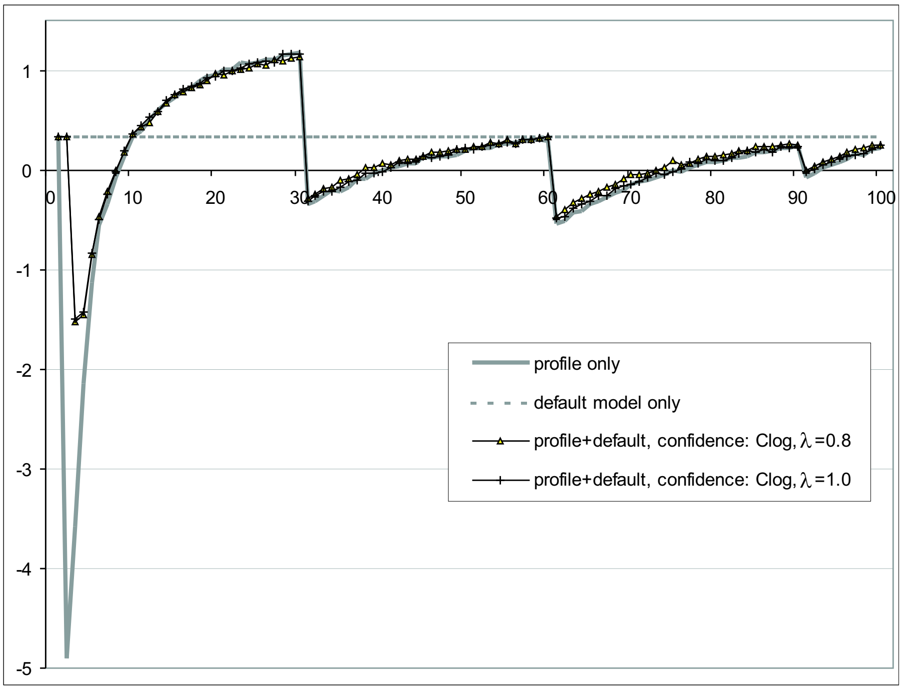


Figure 8.8: Hybrid agents vs. single-model agents: payoffs against "stepping" users, combining strategies, $C^{*\,0.9}_{log}$ used as the confidence measure.

As the possible rewards span a relatively large interval (we are using the payoff table from Figure 8.1 again), the initial temperature parameter is relatively high: $T_0 = 100$, and the decay factor is 0.8. Thus $T_i = T_0 * (0.8)^i$. The results in Figure 8.7 show that the double-model agent has some problems with efficient exploration – in consequence, he plays *too* safe against a static user. On the other hand, he is much better protected from sudden changes in the user's behavior. Moreover, the double-model agent plays much better against a "cheater": he loses 86.9 less than the profile-based agent in the first 15 steps (after that both agents fare almost the same).

8.2.4 Combining Strategies for Games with No Pure Equilibrium

Let us go back to section 8.2.2 and to the assumption that the user's response does not depend on the actual action from the agent. Note that the assumption makes perfect sense when the user simply cannot know the agent's action in advance. This is the case, for instance, when the negotiation process is longer and consists of multiple steps, or when some hidden policy of the bank is concerned (instead of particular "offers"). The game is a matrix game then, and the default strategy pair `(safe offer,cheat)` is the maxmin equilibrium (von Neumann and Morgenstern, 1944).

All the results had been obtained for combining *evaluations* so far. Note that – if we treat the agents' strategies as mixed ones – we can combine them directly (as proposed in Section 6.3.3). This did not prove successful in this setting, as Figure 8.8 shows. It seems that either multilevel combinations of strategies are not suitable for the "banking game", or a different confidence measure should be used in this case.

The games we have been dealing with up to this point, however, are somewhat special since they have their equilibria within the set of pure strategies (i.e. deterministic decisions of the agents). For most matrix games this is not the case. On the other hand, every game has its maxmin equilibrium within the set of *mixed* strategies (von Neumann and Morgenstern, 1944). Consider, for example, the game with the strategic form presented below. The agent's maxmin strategy for this game is $s_{default} = [0.4, 0.4, 0.2]$. If any of the players plays his maxmin, the expected output of the game is 0.

	b1	b2	b3
a1	-1	2	0
a2	0	-2	5
a3	2	0	-10

Remark 8.2 *The sets of all mixed strategies are obviously infinite for both agents. However, only a finite subset really matters while the agent is making his decision: if agent A can guess the opponent's current (mixed) strategy approximately and the strategy is different from the opponent's maxmin, then there is a pure strategy for A that is no worse than any other (mixed) strategy (in terms of the expected payoff). If this is not the case, then A's maxmin strategy provides the best* guaranteed *expected payoff. Thus, each player should consider only his pure strategies* plus *the maxmin strategy while looking for the best decision.*

In the last series of experiments the hybrid agent has been choosing strategy $S = C\ s_{profile} + (1 - C)\ s_{default}$, where $s_{profile}$ is the strategy with the best estimated payoff. A modified confidence measure was used: the confidence was defined as

$$C' = \begin{cases} \min(C^{\Lambda}_{log}, C_{Wang}) & \text{if } \min(C^{\Lambda}_{log}, C_{Wang}) \leq 0.4 \\ \max(0.4, 3\min(C^{\Lambda}_{log}, C_{Wang}) - 1.9) & \text{otherwise} \end{cases}$$

The results (Figure 8.9) reveal that the hybrid agent is again too cautious when the user is random and static. However, the bottom line in the game is drawn by a user who can guess the agent's current strategy S somehow (he must be a 2-level agent rather than 0-level, since the banking agent is a 1-level one). The "malicious" user here is defined this way: he uses a random policy for the first 30 steps, and after that starts choosing the most dangerous action (the one with the minimal payoff), "guessing" the agent's strategy in advance. Playing against a user who chooses the most dangerous action all the time, the hybrid agent was 93.6 better off than the profile-based agent after the first 50 steps.

Remark 8.3 *Note that even the "malicious" 2-level opponent has no way of guessing the banking agent's final* action*, because the agent uses mixed strategies that imply some degree of randomness. Thus, all the opponent can guess is the actual (mixed)* strategy *of our agent.*

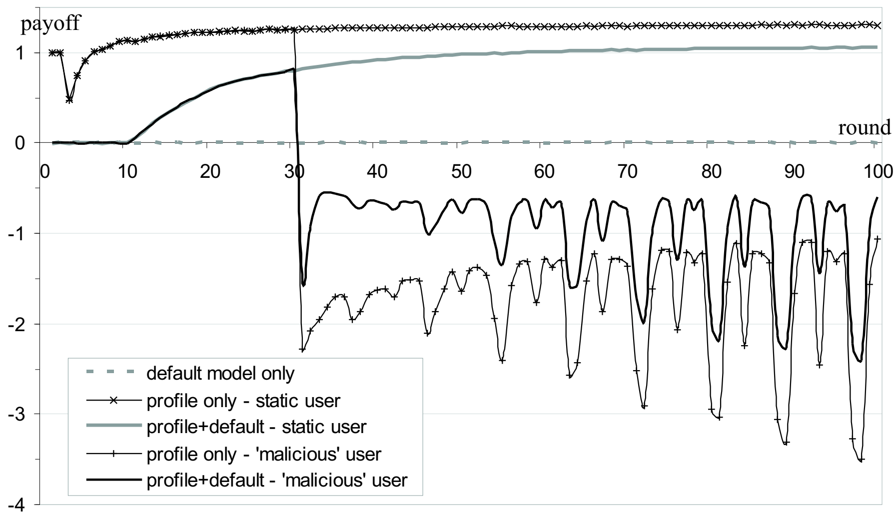


Figure 8.9: Results for the matrix game: static user, "malicious" user

In the next section, multilevel decision making is considered in more sophisticated scenarios, too.

8.3 Multilevel Modeling of Reality with ATL Models

This section presents a preliminary view to hierarchical modeling of the reality, and multilevel decision making, in some more complex scenarios. In most real-life scenarios – unlike the e-banking game we used heavily in the last chapters – the domain of action includes more than one state: there are many possible situations in which the agents' behavior and the response from the environment can be completely different. Moreover, the situations (states) and agents' actions (choices) usually form a complex temporal and strategic structure, showing which actions – and in what way – may lead from one state to another. Multi-player game models seem a natural choice as a representation language for such domains. They are natural and intuitive, they allow to define explicitly the outcome of agents' behavior over time, and they can be extended to include other aspects of multi-agent systems if necessary: uncertainty, knowledge, requirements etc. (cf. Chapters 3, 4 and 5). Last but not least, we can use ATL formulae to specify agents' goals, and use ATL model checking in the way presented in Section 2.8 to find a secure plan that satisfies a goal (if such a plan exists).

Unfortunately, using more complex models demands that the decision-making agent must possess more detailed knowledge as well. For instance, in order to predict opponents' behavior, the agent should know how the opponents are supposed to act *in every particular state*. The purpose of multilevel decision making is to exploit the knowledge in a smarter way. However, to exploit the knowledge, we must first obtain it.

Section 8.2.3 showed that this is hard even in the extremely simple case when there are only three different situations to be considered (there is only one state and three different offers from the banking agent, upon which the user's decisions may depend). The hybrid agents from that section had substantial problems with acquiring an accurate model of the reality. Using the multilevel decision making, the agents played it safer – and when an agent plays safer, his exploration of all the possibilities is less efficient.

Models presented in this section include not three, but dozens to hundreds of situations, for which the behavior of other agents should be somehow predicted. Moreover, simple exploration strategies like the Boltzmann strategy (Banerjee et al., 2000) can have disastrous effects, because the consequences of some actions can be irreversible. For instance, if agent x tries to cooperate with agent y in Example 8.1, he may *never* be able to receive *any* payoff if the other agent turns out to be hostile or irresponsible: first agent y helps x to load the heavy (and more expensive) cargo into the rocket, and then abandons him completely, so that x cannot unload the cargo and replace it with something easier to handle. In such a case, agent x gains knowledge about the other agent, but he will have no opportunity to use it any more. Collaborative modeling (Zukerman and Albrecht, 2001) may prove helpful in such a setting, and the issue is certainly worth further investigation, but we feel that an exhaustive answer to this question is beyond the scope of the thesis. In consequence, we strive to show how multilevel decision making *can* improve agents performance. We do not definitely say if it really does, though. We just present a few example scenarios in which both the safe (maxmin) strategy and an adaptive strategy exploiting some knowledge about the environment of action have obvious advantages, and in which some tradeoff between both approaches looks intuitively appealing.

8.3.1 Examples: Rockets and Bridge

Example 8.1 First, we present a variant of the Rocket Domain example from Section 2.8.3, in which two types of cargo are considered. Again, there are three agents. Agent x – as before – can decide to load a piece of cargo if the rocket is empty (action *load1* for cargo type 1 and *load2* for cargo type 2), unload the cargo from the rocket if the cargo is inside (action *unload*), or move the rocket (*move*); he can also stay passive for a while (action *nop*). This time, however, agent y can only lend a hand (*help*) or do nothing (*nop*), and agent z can only supply the rocket with fuel (action *fuel*) or remain passive (*nop*).

Agents x and z form a team whose task is to deliver the cargo from London to Paris, using the available rocket (propositions atRL and atRP mark the situations in which the rocket is at the London airport and Paris airport, respectively). The game is repeatable now: as soon as x and z deliver the cargo to Paris, they get paid and the cargo is taken away from the airport – and the agents are free to go back and pick up another piece. The more they deliver, the more they earn, of course. The states in which a piece of cargo lies inside the rocket are labeled with propositions $\mathsf{inC1}$ for cargo type 1 and $\mathsf{inC2}$ for cargo type 2, respectively. The London airport has infinite supplies of both types of cargo, so there is no need to indicate the presence of cargo in London with a special proposition any more (it would simply hold in every state).

The first type of cargo is easier to handle: x can load and unload it on his own.

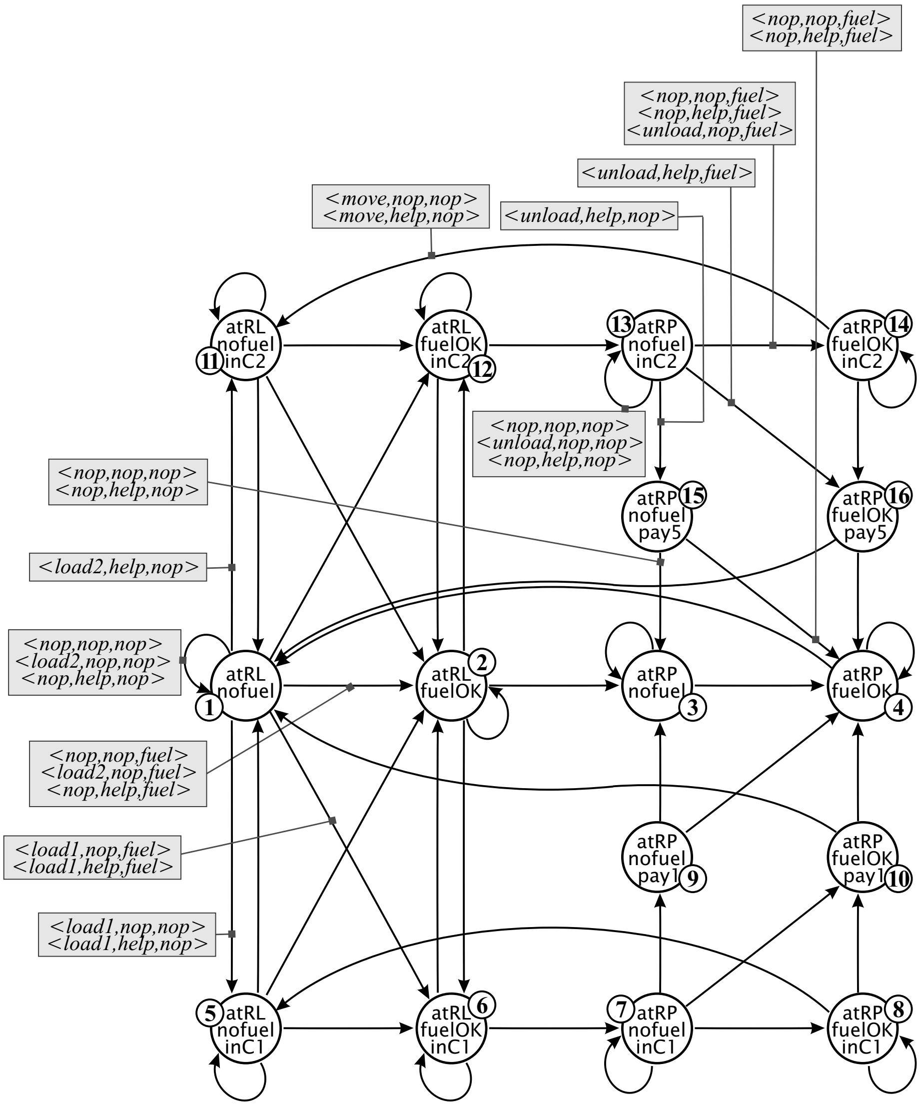


Figure 8.10: Rocket domain again: a multi-player game model for agents who deliver the cargo for money

Cargo type 2 can be loaded/unloaded only with the help from y. On the other hand, cargo type 1 is less profitable: the agents earn on it only 1 acur (Agent CURrency unit) per delivery, while the second type of cargo is worth 5 acurs when delivered to Paris (propositions pay1 and pay5). The temporal and strategic structure of the system is shown in Figure 8.10.

Essentially, x and z can try two sensible strategies: they can try to cooperate with y in shipping the more valuable cargo, or assume that y may be unwilling (or unable) to cooperate, and play the game safe (i.e. stick to the cheaper cargo). The safe strategy can be computed through the "planning as model checking" procedure:

$$\begin{aligned} plan(\langle\langle x,z\rangle\rangle\Diamond \mathsf{pay5}) &= \{\ \langle 15,-\rangle, \langle 16,-\rangle\ \} \\ plan(\langle\langle x,z\rangle\rangle\Diamond \mathsf{pay1}) &= \{\ \langle 1, x\!:\!load1\cdot z\!:\!fuel\rangle, \langle 2, x\!:\!load1\cdot z\!:\!nop\rangle, \\ &\qquad \langle 3, x\!:\!nop\cdot z\!:\!fuel\rangle, \langle 4, x\!:\!move\cdot z\!:\!nop\rangle, \\ &\qquad \langle 5, x\!:\!nop\cdot z\!:\!fuel\rangle, \langle 6, x\!:\!move\cdot z\!:\!nop\rangle, \\ &\qquad \langle 7, x\!:\!unload\cdot z\!:\!nop\rangle, \langle 8, x\!:\!unload\cdot z\!:\!nop\rangle, \\ &\qquad \langle 9,-\rangle, \langle 10,-\rangle, \\ &\qquad \langle 15, x\!:\!nop\cdot z\!:\!nop\rangle, \langle 16, x\!:\!nop\cdot z\!:\!nop\rangle\ \} \end{aligned}$$

Thus, x and z have no infallible plan to deliver cargo type 2 to Paris – unless they have already done so. On the other hand, they can manage the cheaper cargo on their own if they never try the more expensive one: they have a collective strategy to eventually enforce $\Diamond$pay1 from every state except 11, 12, 13 and 14 (we can also express it with the ATL formula $\neg \mathsf{inC2} \rightarrow \langle\langle x,y\rangle\rangle\Diamond \mathsf{pay1}$). Therefore the strategy with the highest guaranteed outcome (the "safe" strategy) is:

$$\begin{aligned} \{\quad &\langle 1, x\!:\!load1\cdot z\!:\!fuel\rangle, \langle 2, x\!:\!load1\cdot z\!:\!nop\rangle, \\ &\langle 3, x\!:\!nop\cdot z\!:\!fuel\rangle, \langle 4, x\!:\!move\cdot z\!:\!nop\rangle, \\ &\langle 5, x\!:\!nop\cdot z\!:\!fuel\rangle, \langle 6, x\!:\!move\cdot z\!:\!nop\rangle, \\ &\langle 7, x\!:\!unload\cdot z\!:\!nop\rangle, \langle 8, x\!:\!unload\cdot z\!:\!nop\rangle, \\ &\langle 9,-\rangle, \langle 10,-\rangle, \langle 15,-\rangle, \langle 16,-\rangle \quad \} \end{aligned}$$

Intuitively, the other strategy also makes sense: x and z presumably do not know the attitude and/or preferences of y at the beginning. If y is friendly and decides to help, they can achieve pay5; the only way to try it is to execute $x : load2 \cdot z : nop$ or $x\!:\!load2\cdot z\!:\!fuel$ at state 1, and see what y does. It is worth pointing out, though, that such a strategy can have disastrous effects if y turns out to be hostile. First, x and z have no way of guaranteeing pay5 on their own. What is even more important, when they try to cooperate with y, they leave themselves vulnerable: if y helps to load the cargo, and then refuses to unload it, they will *never* receive any payoff.

On the other hand, x and z may have some information about y that makes them believe that y is rather likely to cooperate. In such a case, they should probably try to establish the cooperation – if only they are sufficiently confident in this belief. Figure 8.11 shows an example of such a situation: the team keeps a probabilistic model

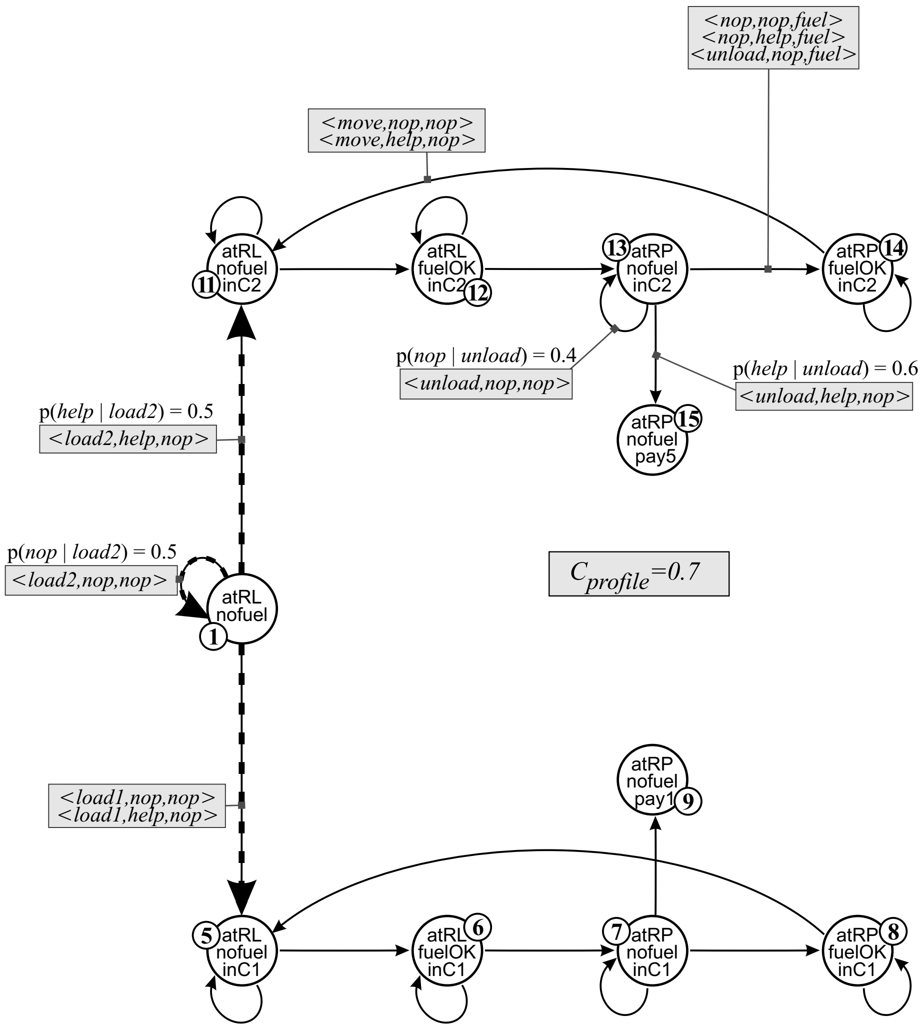


Figure 8.11: Multilevel decision making in the rocket domain: should we play safe or try to cooperate?

of y's behavior. For instance, they expect that y helps to load the cargo type 2 in London with probability $p(help|load2) = 0.5$, and the probability that y refuses as $p(nop|load2) = 0.5$. They also believe that y is rather likely to help with unloading the cargo in Paris: $p(help|unload) = 0.6$, $p(nop|unload) = 0.4$, and they are confident in $C_{\mathit{profile}} = 0.7$ that the model is accurate. The team assumes that the actions of y are deterministic, given the state and choices from other agents; in other words, x and z hold probabilistic beliefs about which deterministic policy of y is really the case.

The team considers two strategies: the agents can try to deliver a piece of cargo type 1:

$$\begin{aligned} S_1 &= \{\langle 1, x\!:\!load1\cdot z\!:\!fuel\rangle, \langle 5, x\!:\!nop\cdot z\!:\!fuel\rangle, \langle 6, x\!:\!move\cdot z\!:\!nop\rangle, \\ &\quad \langle 7, x\!:\!unload\cdot z\!:\!nop\rangle\}, \end{aligned}$$

or a piece of cargo type 2:

$$\begin{aligned} S_2 &= \{\langle 1, x\!:\!load2\cdot z\!:\!fuel\rangle, \langle 11, x\!:\!nop\cdot z\!:\!fuel\rangle, \langle 12, x\!:\!move\cdot z\!:\!nop\rangle, \\ &\quad \langle 13, x\!:\!unload\cdot z\!:\!nop\rangle\}. \end{aligned}$$

The strategies are shown in Figure 8.11, too.[3] To evaluate S_2, we first observe that the expected payoff for this strategy in state 11 (according to the profile of y kept by the team) is: $0.6 \cdot 5 + 0.4 \cdot 0 = 3$. Thus,

$$E(\mathit{profile}, S_2) = 0.5 \cdot 3 + 0.5 \cdot 1 = 2.$$

Moreover, $E(maxmin, S_2) = 0$ because, if y turns out to be hostile, the team gets no payoff. The multilevel evaluation of S_2 is thus:

$$eval(S_2) = C_{\mathit{profile}} \cdot E(\mathit{profile}, S_2) + (1 - C_{\mathit{profile}}) \cdot E(maxmin, S_2) = 1.4$$

At the same time, $eval(S_1) = 1$ (because S_1 brings always 1 acur, regardless of the response from y), so the team should seriously consider an attempt at the cooperation with y. □

Example 8.1 presented a situation in which the planning agents did not know the preferences of the opponent. The following example demonstrates that using a model of the opponents can be beneficial even in a zero-sum game, when the other agents are obviously enemies: the playing agent may try to exploit potential mistakes of the enemies if they are likely to choose suboptimal strategies. Moreover, if we have a model of the enemy, but we are uncertain about its accurateness, some balance between exploiting the model and using the maxmin strategy can be a good solution.

Example 8.2 A bridge game is being played: we consider only the final part of the card play, in which the deck consists of only 16 cards. There are three active players at this stage: one player controls the cards at positions N and S (we call the player NS), and his enemies (W and E) control the hands at positions W and E, respectively. At this moment we assume that all the players can see each others' hands: they play a perfect

[3] The graph and the presented strategies include only the relevant states, transitions and choices.

information game. NS starts from the hand labeled N. A fragment of the game tree is shown in Figure 8.12. NS can open his play with $A\heartsuit$ or any of the spades (it does not really matter which one). If he plays a spade, then E and W must join with $A\spadesuit$ and $K\spadesuit$ respectively (players are obliged to follow suit unless they are void in it), and E wins the trick for his team. If NS opens with $A\heartsuit$, E can play any of his cards ($A\spadesuit$, $K\clubsuit$, $Q\clubsuit$ or $3\clubsuit$), then NS can drop any card from the S position, and W completes the trick with $K\heartsuit$, $Q\heartsuit$ or $J\heartsuit$; NS wins the trick in this case.

For "bridge heads", we describe the setting of the game. Contract bridge with tournament scoring is played; the contract being played is 1NTx (1 No Trump doubled), and both parties are vulnerable. Up to this point, NS has taken 4 tricks, and the team of W and E 5 tricks. NS must collect the total of 7 tricks (out of 13) to make the contract; thus, he still needs 3 more tricks (out of 4 remaining). If he makes the contract, he earns 130 points (together with his partner). One overtrick (i.e. taking all the remaining tricks) gives additional 200 points; if the contract is down, NS pay the penalty of 200 points for the first trick below the threshold, and 300 for each subsequent one. For most readers, the list of possible payoffs for NS is the only important factor in this example. The list is given below; the team WE gets the opposite value:

- if NS take all 4 remaining tricks, they get 330 points;
- if NS take 3 tricks, they get 130 points;
- if NS take 2 tricks, they get -200 points (i.e. they lose 200 points);
- if NS take 1 tricks, they get -500 points;
- if NS take no tricks, they get -800 points.

A game model for the game is sketched in Figure 8.13. Only relevant paths are included (it makes no difference, for instance, whether NS plays $Q\spadesuit$, $J\spadesuit$ or $3\spadesuit$ at the beginning). Payoffs are represented with propositions $\mathsf{pay(n)}$. The model is turn-based, i.e. at every state only one player effects the next transition (other agents are assumed to choose the "do nothing" action). Each transition in the graph is thus labeled with the name of the player who is active at the moment, and his actual choice that executes this transition.

The thick arrows show the maxmin strategy of NS. Note that, if NS uses this strategy, he gets -200 points, no matter what the other players do. Note also that opening with $A\heartsuit$ leads to the payoff of -500 points if the enemies choose the optimal defense: E throws away his Ace, and W takes the remaining 3 tricks in consequence. However, E must make a conscious effort and think the game over carefully to throw away his $A\spadesuit$ instead of $3\clubsuit$ – most players usually follow the reflex to do the reverse.[4] If E follows the reflex and keeps his Ace, NS will inevitably win the game and earn $+130$. Thus, if NS suspects that E is a novice or that he usually plays automatically with no careful analysis of the deal, the risky strategy of playing $A\heartsuit$ first can be worth trying.

[4] Actually, this example has been inspired by a game that occured during a high level tournament. The game was designed to have a completely safe winning strategy, yet many players did not see it – so strong is the automatic reflex not to throw away one's high cards in favor of the lower ranks (Macieszczak and Mikke, 1987).

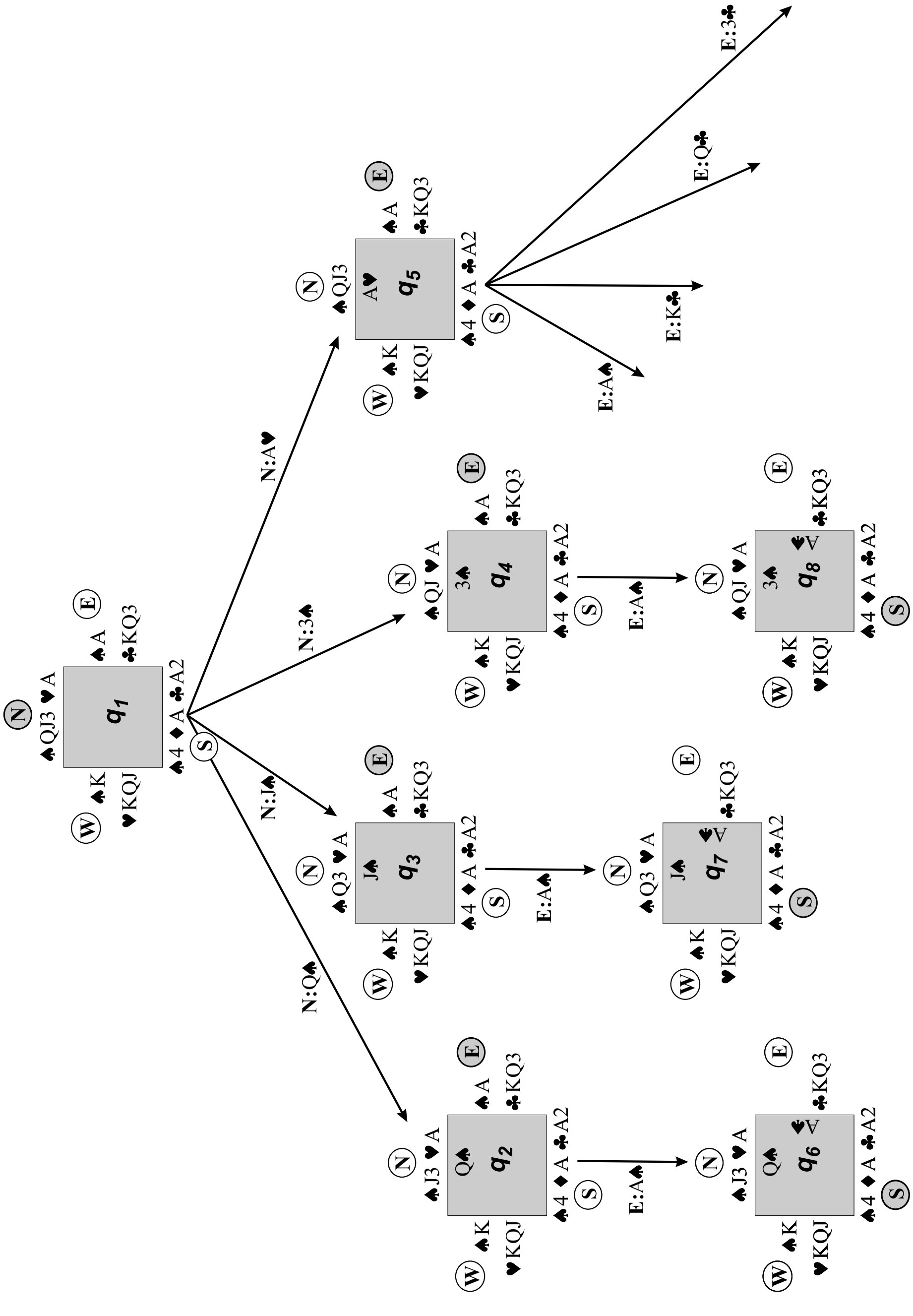


Figure 8.12: The bridge game: a fragment of the game tree (hint: cock your head to the left in order to read the picture more easily)

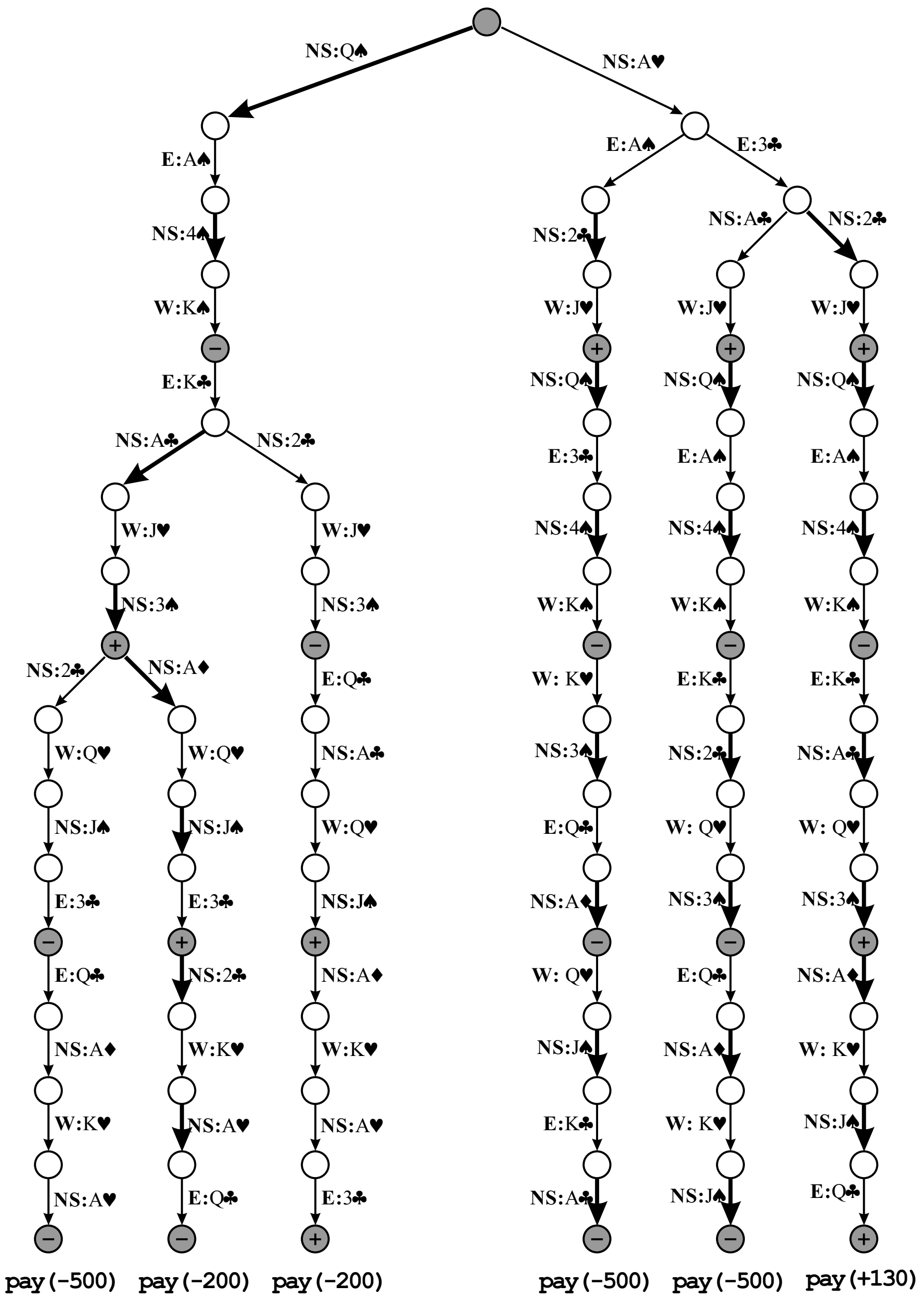


Figure 8.13: The bridge game: game model. Only the relevant paths are included. The "grey" states mark the end of a trick. The "+" sign indicates that the trick has been won by NS; the "–" sign tells that the trick has been lost by NS (i.e., won by the coalition of W and E.

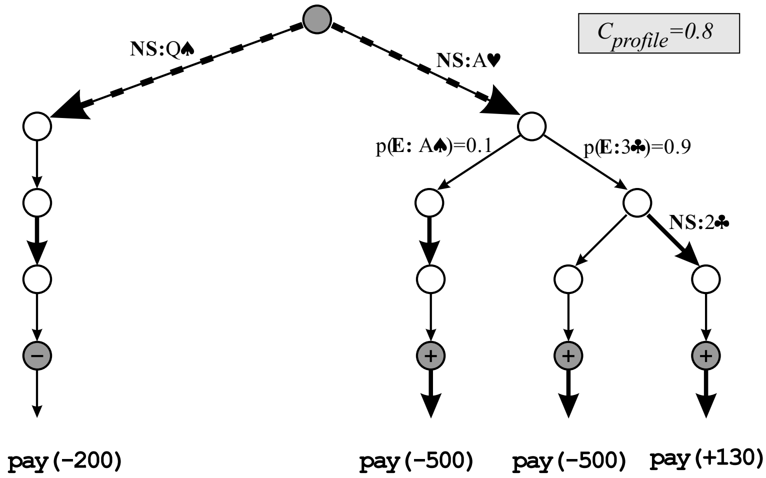


Figure 8.14: The bridge game: exploiting predicted mistakes of the opponents.

Suppose that NS estimates the probability that E will actually make the mistake and throw away 3♣ as $p(E : 3♣) = 0.9$, and the probability of the optimal choice from E as $p(E : A♠) = 0.1$; at the same time, NS estimates his self-confidence as $C_{\mathit{profile}} = 0.85$ (see Figure 8.14). Let S_1 be the NS's maxmin strategy, and S_2 the strategy obtained from S_1 through changing the first move from $NS : Q♠$ to $NS : A♡$. The maxmin strategy gives always -200 points, so $eval(S_1) = -200$. With the other strategy, we obtain

$$\begin{aligned} E(\mathit{profile}, S_2) &= 0.1 \cdot (-500) + 0.9 \cdot 130 = -50 + 170 = 67 \\ E(maxmin, S_2) &= -500 \end{aligned}$$

and

$$\begin{aligned} eval(S_2) &= C_{\mathit{profile}} \cdot E(\mathit{profile}, S_2) + (1 - C_{\mathit{profile}}) \cdot E(maxmin, S_2) \\ &= 0.8 \cdot 67 + 0.2 \cdot (-500) = 53.6 - 100 \\ &= -46.4 \end{aligned}$$

Again, NS should probably try the risky strategy to get a better payoff. □

ATL (and its models) can be extended along various dimensions to include notions like knowledge or requirements. We use the two final short examples to hint the potential that ATEL (or ATOL) and DATL offer for the analysis of more complex scenarios. Obviously, we may employ multi-player epistemic game models and ATOL formulae to describe the natural bridge setting: when the players do not see each others' hands. DATL can be used, for instance, to include cheating in the scope of the model, and ask whether such an undesirable behavior can be prevented – or specify sanctions that should occur when it has been detected.

Example 8.3 Players seldom have complete knowledge about the distribution of cards in a bridge game. Every player sees only his own hand and the "dummy" – the cards at the position of N in this case. Figure 8.15 presents a fragment of the game tree with epistemic accessibility relations for our 16-cards bridge problem in this more natural setting: the players do not see each others' hands any more.

The ATOL formula $\langle\langle NS\rangle\rangle^{\bullet}_{Obs(NS)}\Diamond\mathsf{pay(+130)}$ defines the "maximal" goal for NS; unfortunately, $q_1 \not\models \langle\langle NS\rangle\rangle^{\bullet}_{Obs(NS)}\Diamond\mathsf{pay(+130)}$ for the model from Figure 8.15. NS has a uniform strategy to get -200 points in the game: $q_1 \models \langle\langle NS\rangle\rangle^{\bullet}_{CO(\varnothing)}\Diamond\mathsf{pay(-200)}$, but he is not able to identify the successful strategy ($q_1 \not\models \langle\langle NS\rangle\rangle^{\bullet}_{Obs(NS)}\Diamond\mathsf{pay(-200)}$), and he is not even aware that it exists at all ($q_1 \not\models K_a\langle\langle NS\rangle\rangle^{\bullet}_{CO(\varnothing)}\Diamond\mathsf{pay(-200)}$). Only the payoff of -500 can be guaranteed ($q_1 \models \langle\langle NS\rangle\rangle^{\bullet}_{Obs(NS)}\Diamond\mathsf{pay(-500)}$): NS can for instance play 3♠ and wait until the opponent that wins the trick leads in ♡ or ♢. Note that -500 points is not very much in this game: in fact, NS has no strategy to make *less*. Thus, if NS has some probabilistic model of the actual distribution of cards, and possible responses from the opponents, he can benefit from using it in a similar way as in Example 8.2. □

Game theory is sometimes criticized because it treats idealizations rather than the real phenomena it was supposed to tackle. In a sense, game rules are treated too seriously in game theory: when the rules say that players must follow suit, we usually do not even include a possibility that some players may not obey the rule in the model of the game (see Figure 8.14, for instance). Hence, game theory models seem hard to use when we want to induce emergent behavior, since emergence of new ideas often comes through re-defining the *structure* of the existing model of the reality (or putting it in another way, through re-defining the search space). Moreover, game-theoretical analysis usually omits important phenomena like cheating, illegal epistemic updates (overlooking, eavesdropping) etc., that occur in real games, but not in their game-theoretical models.

Game rules are unlike the laws of physics: they can be broken. The chapter on Deontic ATL shows how we can include illegal (albeit "physically" possible) situations in our models, and at the same tell them apart from the legal ones with the deontic accessibility relations.

Example 8.4 Figure 8.16 shows how the possibility of cheating can be included in the game model: some players can replace a card of theirs with one they hold in their sleeve; also, W and E can try to swap some cards of theirs or some player can try to play his card at somebody else's turn. Deontic relations are used to deem illegal every situation that results from a change of the current distribution, other than playing a card according to the rules of bridge.

Note that the requirements are local in the model: state q_2 should be considered legal when q_1 is the initial situation, but it cannot be obtained in a legal way when we start from q_1'. On the other hand, q_2' is an acceptable alternative of q_1', whereas q_2 cannot be legally achieved when we start from distribution q_1'. Note also that the model specifies legality of states rather than actions (cf. Section 5.2.1). In particular, attempted (but unsuccessful) dishonest behavior cannot be defined as illegal in this model: it is only the result of agents' actions that matters in this respect. However, the

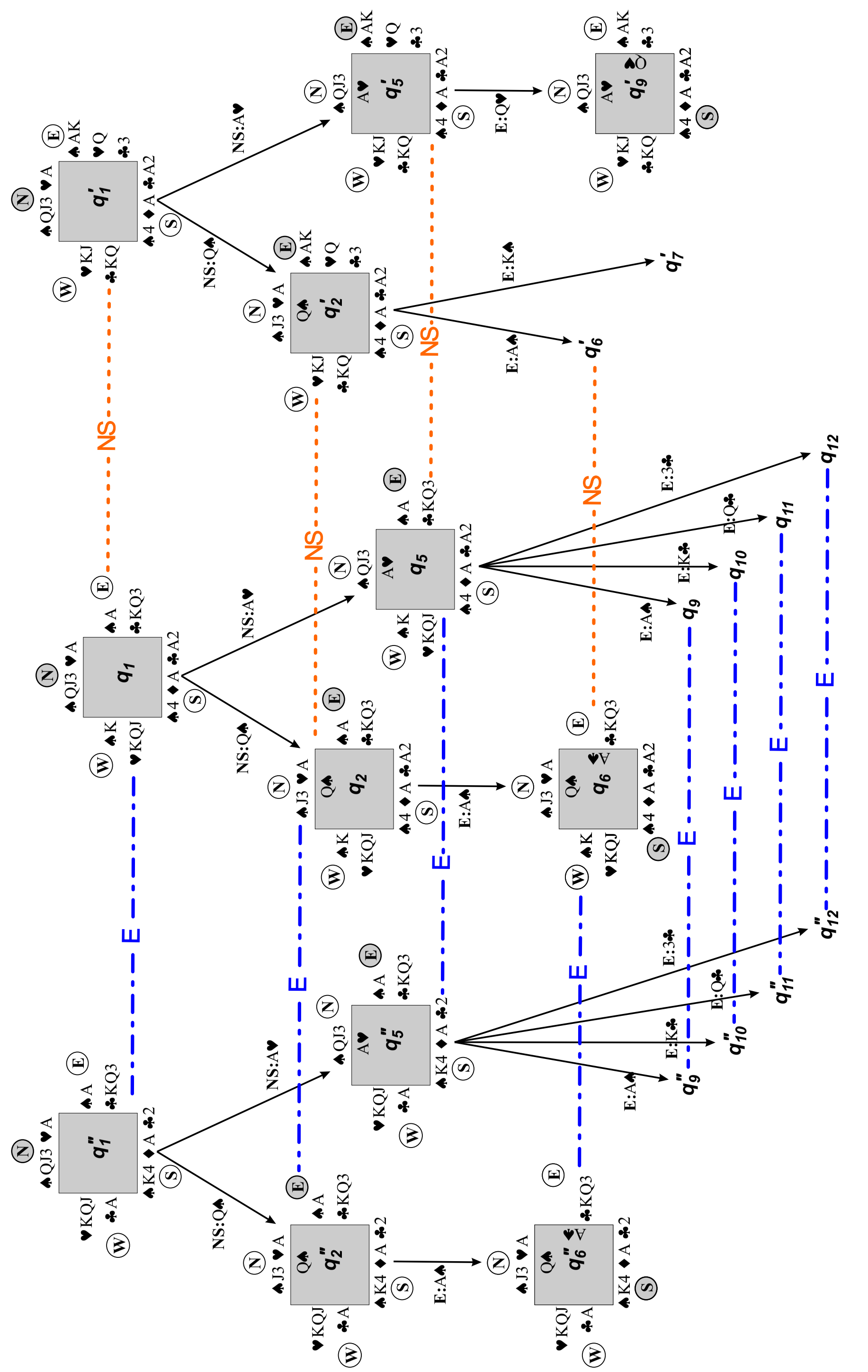


Figure 8.15: The bridge game: players do not see each others' cards any more. The dotted lines display epistemic accessibility relations; reflexive epistemic links are not shown for the clarity of the picture.

model can be enhanced to include a "trace" of the most recent agents' choices in the states – in the way we did it in Example 4.8 – and then we are free to mark every tuple of choices as legal or not by determining the acceptability of the resulting state.

Looking for a safe plan with maximal (guaranteed) outcome, agent NS may now check formulae

$$\langle\langle NS \rangle\rangle \Diamond \mathsf{pay(n)}$$

in the initial state of the model (let it be q_1), for every possible payoff value n. However, if the agent wants to be honest, he should rather look for a strategy that allows him to *properly enforce* the best pay(n) in q_1, i.e. he must check the following family of properties:

"given that $q_1 \models \mathcal{O}\varphi \wedge \mathcal{UP}\varphi$, we have $q_1 \models \langle\langle NS \rangle\rangle \varphi \mathcal{U} (\varphi \wedge \mathsf{pay(n)})$,"

and choose the plan that is successful for maximal n.

The agent can have a more specific model of the opponents, too, that gives some prediction of how likely various actions of the opponents are – including illegal actions as well (see Figure 8.16 again). Obviously, such a profile of the opponents can be exploited as before. NS can evaluate available strategies according to the profile, and both resulting evaluations – $E(maxmin, S)$ and $E(\mathit{profile}, S)$ – can be combined in the same way as in Example 8.2. □

8.3.2 Concluding Remarks

In this chapter, we tried to show how the multi-level decision making can help to bridge the gap between normative, game theory-based models of the reality, and adaptive models obtained through machine learning, collaborative modeling etc. Normative models usually assume some boundary shape of the environment, while adaptive models try to approximate the actual shape of it – and both have their advantages for an agent who tries to make good decisions under uncertainty. Using a linear combination of both kinds of model should make the agent play relatively safe, while being (relatively) adaptive at the same time. Results of the experiments from Section 8.2 support the hypothesis to some extent.

On the other hand, they also show that several pieces of the puzzle are still missing if we want to use the decision making scheme effectively. A good confidence measure is necessary; we may also need a better exploration-exploitation scheme. Most importantly, an efficient way of acquiring a "profile" of the reality is crucial here. This seems most evident in the context of Section 8.3: the examples include hundreds of states, and some courses of action can be costly or even irreversible. Computing such a large (adaptive) model of the reality (i.e. determining all the weights, probabilities etc.) via straightforward Q-learning or Bayesian updating, for instance, does not seem feasible at all. One way that may help to overcome this problem is to reduce the state space by simplifying the characteristic of the domain. Collaborative modeling, where a model of an agent's behavior in a particular state can be "transferred" from models for agents or states we consider similar, and for which we already obtained a good predictor, seems also promising. As both ideas comprise a non-trivial task, we do not pursue them in this thesis, leaving them rather for future research.

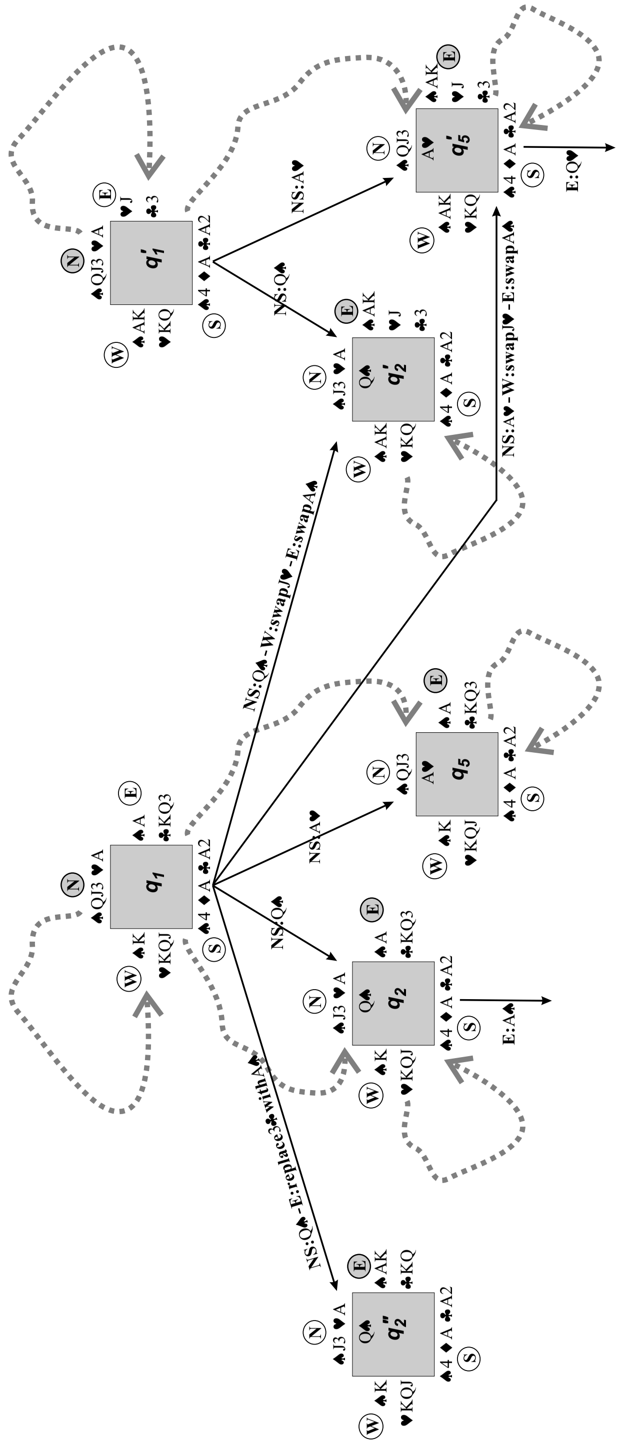


Figure 8.16: The bridge game: honest play and cheating in one box.

Chapter 9

Conclusions

SYNOPSIS. *The story draws to its end. Models of reality, modal logics, confidence measures, and planning algorithms have made their appearances in this book so far. We considered servers and clients, card games, trains, tunnels, malicious users, and simple e-banking agents. Even James Bond dropped in for a moment to show us how we can save the world when it is really required. Now the time comes to conclude that the show was worthwhile. And, of course, there are fair chances for a sequel.*

9.1 A Short Look Backwards

This thesis considers several aspect of agents and their communities. As we explained in Chapter 1, we primarily see *multi-agent systems* as a metaphor for thinking and talking about the world, and assigning it a specific conceptual structure. This comes suspiciously close to our view of logic-based approaches to Artificial Intelligence and Cognitive Science. The main appeal of formal logics lies – in our opinion – in the fact that it provides us with a vocabulary to talk and think about reality. The vocabulary is precise and demands precision when we use it to define conceptual structures that are meant to model the phenomena we want to investigate. We are also provided a methodology that enables to investigate the consequences of our conceptual choices in a systematic way. The first part of the thesis presents thus an attempt to study multi-agent systems through suitable logics and their models, in the hope that this can induce some informal understanding of the phenomenon as well.

We chose Alternating-time Temporal Logic (Alur et al., 2002) as the basis for our studies, because it builds on a number of notions we consider crucial for multi-agent systems. On one hand, it refers to the game theory concepts of agents, teams, actions, choices, strategies and their outcomes; on the other hand, the temporal layer of ATL allows one to refer to situations and their changes, and to alternative courses of action. The formal investigation of ATL produced equivalence results for the different semantics of this language, and proved that Coalition Logic (Pauly, 2001a) – another recent logic of strategic ability – is strictly subsumed by ATL. However, we tried also to show ATL and its extensions as a part of a bigger picture: tracing its inspirations, and pointing out similarities to other languages. Moreover, the extended languages themselves: ATEL, ATOL, DATL etc. show that ATL can be seen as a generic framework for multi-agent systems, which we can augment with additional concepts when we need them.

Studies on Alternating-time Temporal Epistemic Logic ATEL (van der Hoek and Wooldridge, 2002) proved especially insightful. On one hand, ATEL provides our models with the notion of (qualitative) uncertainty, and tools to describe agents' knowledge about the actual situation – and it is hard to imagine a good representation of agents that neglects these issues. On the other hand, the strategic and epistemic layers in ATEL are combined as if they were independent. They are – if we do not ask whether the agents in question are able to identify and execute their strategies. They are not if we want to interpret strategies as *feasible plans* that guarantee achieving the goal. Intuitively, strategic abilities of an agent should *heavily* depend on his knowledge – it looks very simple when we put it like this. But it took much time and a few failed attempts before we arrived at the right conclusions, and proposed ATOL and ATEL-R as remedies.

After devoting much of the thesis for studies on existing logics for agents, we proposed a new extension of ATL ourselves. DATL extends ATL with the deontic notion of obligation. This time, the strategic and deontic layers seem to be *really* orthogonal. Is that so? Let us leave it as a challenge for the new, forthcoming wave of avid young researchers.

As repeatedly stated throughout the thesis, we tried to focus on models, and the way they (together with the semantic rules) reflect our intuitions about agents acting in multi-agent environments. *Decision making* was one of the most prominent issues here. Apart of investigating cooperation modalities and their semantics as the formal counterpart of the decision making process, we devised an efficient procedure for multi-agent planning to achieve goals that can be expressed with ATL formulae. We also proposed a satisfiability preserving interpretation of ATEL, BDI_{CTL} and DATL into ATL, thus extending the scope of the planning algorithm to goals that include epistemic and deontic properties as well. Unfortunately, ATOL and ATEL-R turned out to be intractable. Thus, such an efficient procedure does not exist in these cases.

The theme of decision making provided a link to the second part of the thesis, in which a concept of multi-level modeling of reality was proposed, where various models of the same environment could be combined to improve decision making. Our main motivation was to try to endow agents with some way of using secure game theory-based models (ATL models, for instance) together with adaptive models obtained through machine learning. After trying several confidence measures, the one based on logarithmic loss function produced some promising results – and, finally, experiments with simulated software agents showed that indeed the agents can try to be generally adaptive and relatively secure at the same time.

So, what have we learned about agents? Did we enhance our "informal understanding" of the way they act in the world and make their decisions? It is hard to say it with logic, and ambiguous to say it without. One thing is certain: more and more research paths can be seen, open and unexplored yet.

9.2 Into the Future

Further research on ATL and its cousins and extensions can be pursued along at least three different dimensions:

- the theoretical dimension:
  - meta-properties of the logical systems – expressive power, decidability, axiomatizability, complexity of model checking etc.
  - relationships between various logics and their models – equivalence, subsumption, semantic interpretations and reductions of one logic into another;
- the philosophical/conceptual dimension:
  - "tailoring" the logics to our intuitions about various aspects of agents and their societies, so that we obtain formal counterparts of our "commonsense" concepts,
  - further formal investigation of the concepts, their interrelationships and interference;
- the practical dimension:
  - application of the techniques developed for the ATL-based logics to seek (possibly suboptimal) solutions of more complex games,
  - modeling real-life systems with models based on modal logics and game theory,
  - planning and decision making for agents in more realistic situations.

A number of specific problems and research questions, that can be seen immediately, is listed below. For some of them, the research is already going on.

1. There are several fundamental concepts in game theory, such as preference relations between outcomes and *Nash equilibria*, that are obviously seem worth investigating in the context of concurrent game structures and alternating-time logics. The first step along these lines of research have already been made by van Otterloo et al. in (van Otterloo, 2004) and (van Otterloo and Jonker, 2004).

2. Coalition effectivity models can be used for logics like ATOL, ATEL-R* and DATL to obtain mathematically elegant semantics. Moreover, relevant equivalence or subsumption results should be established between various semantics for these logics.

3. The parallels between ATEL and BDI_{CTL} suggest that the BDI notions of *desire* and *intention* can enrich ATEL directly, both on the syntactical an semantic level. The issue of how these extensions carry over to ATOL and ATEL-R* can be then investigated, too.

4. ATL and coalition games can provide BDI models with a finer-grained structure of action (simultaneous choices). Furthermore, the cooperation modalities can be "imported" into the BDI framework to enable modeling, specifying and verifying agents' strategic abilities in the context of their beliefs, desires and intentions. The treatment of group epistemics from ATEL can be used in the BDI logics too.

5. The authors of ATL had proposed "ATL with incomplete information" (Alur et al., 2002) before ATEL, ATOL and ATEL-R* were ever defined. It can be interesting to see if this earlier proposal captures the notion of players under uncertainty in a way consistent with intuition – and how "ATL with incomplete information" relates to the epistemic logic-based extensions studied in this thesis. Also, stronger languages like alternating-time μ-calculus can be extended to include agents beliefs under incomplete information, as well as obligations, requirements, desires, intentions etc.

6. In Section 4.3.2, "dynamic logic-like cooperation modalities" were briefly introduced: $[F_A]\Phi$ meaning "A can use strategy F_A to bring about Φ" (or: "every execution of F_A guarantees Φ"). More research on this subject should follow.

7. Model checking of ATL formulae generalizes minimaxing in zero-sum (i.e. strictly competitive) games. It can be interesting to model the non-competitive case within the scope of ATL as well: while checking $\langle\langle A\rangle\rangle\varphi$, the opponents $\mathbb{A}\mathrm{gt} \setminus A$ may be assumed different preferences and/or goals than just to prevent A from achieving φ. Then, assuming optimal play from $\mathbb{A}\mathrm{gt} \setminus A$, we can ask whether A have a strategy to enforce φ provided that $\mathbb{A}\mathrm{gt} \setminus A$ desire ψ. The issue of bringing agents' preferences into the scope of ATL has been addressed recently in (van Otterloo, 2004), but there is still a lot of work to be done.

8. Opponents' preferences are usually used in game theory merely to imply the opponents' strategy under the rationality and optimal defense assumption. Alternatively, we can ask about agents' abilities if the opponents are directly assumed to take a particular line of play. More specifically, we can ask whether A have a strategy to enforce φ provided that $\mathbb{A}\mathrm{gt} \setminus A$ intend *to bring about* ψ, or even to execute a collective strategy $F_{\mathbb{A}\mathrm{gt}\setminus A}$. This can trigger an interesting discussion about the nature of intentions: contrary to the treatment of intention in (Rao and Georgeff, 1991, 1995; Wooldridge, 2000) – where intentions are in fact presented as a special kind of desires (the ones to which the agent is more committed and striving to satisfy them) – we feel that agents intend to *do* (an action) rather than to *be* (in a state). This distinction closely resembles the controversy between action-related and situation-related obligations (cf. Chapter 5). Modal logics of intention might happen to follow the evolution of modal approaches to deontic notions.

9. The link between ATL model checking and minimaxing can be exploited in other ways, too. Efficient pruning techniques exist for classical minimaxing – it may be interesting to transfer them to ATL model checking. Various phenomena, studied in the context of games with incomplete information within game theory, might prove worthwhile to be transfered into the framework of ATL – or rather

ATEL and ATOL: probabilistic outcomes, best defense criteria for games with incomplete information (Frank, 1996; Frank and Basin, 1998; Jamroga, 2001a), non-locality (Frank and Basin, 1998), efficient suboptimal algorithms for games with uncertainty (Frank et al., 1998; Ginsberg, 1999) etc.

10. ATEL seems the largest relevant subset of ATEL-R* with tractable model checking, but this needs to be verified. Also, the conjecture that ATEL formulae provide strongest necessary conditions for their ATOL and "vanilla" ATEL-R* counterparts should be checked.

11. The approximate evaluation of ATOL formulae through their ATEL necessary condition counterparts, suggested in Section 3.4.5, strongly resembles the technique of Monte Carlo Sampling (Corlett and Todd, 1985; Ginsberg, 1999). Similar techniques, like vector minimaxing and payoff-reduction minimaxing (Frank et al., 1998), and generalized vector minimaxing (Jamroga, 2001a) can be tried as well.

12. The "planning as ATL/ATEL model checking" approach should be eventually applied to some real-life problems: for example, to decision making in complex card games like bridge, security analysis, or planning for e-commerce agents. As such domains yield usually huge sets of states, optimization techniques like unbounded model checking (Kacprzak and Penczek, 2004) may be worth testing in such realistic settings.

13. The interpretations presented in Sections 3.4, 3.4.6, 3.5 and 5.4.2 yield immediate model checking reductions for ATEL, ATEL*, BDI$_{CTL}$ and DATL into simpler languages. Our intuition is that the interpretations can be adapted to contexts other than model checking: for instance, to reduction of (general) validity of formulae. Further reduction can also be tried: for example, it may be possible to interpret ATEL and ATL in CTL, or at least CTL+K (i.e. CTL plus knowledge operators).

14. We argued in Chapter 4 that the cooperation modalities should refer to the agents' ability to identify and execute a plan that enforces a property. In order to capture all subtleties of this approach, we introduced a number of families of modal operators. van Otterloo and Jonker (2004) take a different line: they redefine the semantics of existing cooperation modalities, so that – combining them with ordinary epistemic operators – one can express the property of having a strategy *de re*. The research is somewhat preliminary so far, but it looks very promising. Still, it can be interesting to investigate the relationship between the language from (van Otterloo and Jonker, 2004) on one hand, and ATOL and ATEL-R* on the other.

15. Further meta-theoretical analysis of ATOL, ATEL-R* and their subsets (decidability, axiomatizability, decidability of model checking) is possible. Also, complexity of various problems for subsets of ATEL-R* can be investigated.

16. Theoretical properties of our "Deontic ATL", and its relation to other existing systems that combine deontic and temporal/strategic perspective, should be investigated. In particular, a closer study of the relationship between DATL and the "Social Laws for ATL" approach (van der Hoek et al., 2004) – especially the way both approaches can complement each other – should be conducted.

17. In Chapter 5, we demonstrated a number of interesting properties that relate agents' abilities and obligations. However, the properties are defined on the semantic level in the general case of local obligations. It can be interesting to try to express them in the object language as well (it may require some redefinition of the semantics of deontic operators and/or cooperation modalities).

18. In "Deontic ATL", states (or computation paths) form the point of reference for obligations (we name such obligations *requirements*), while obligations can be also understood as referring to agents' actions or even strategies. Technically speaking, this is not a big problem: every model can be enhanced so that each state includes a "trace" of the most recent agents' choices, and then we are free to mark every tuple of choices as legal or not by determining the acceptability of the resulting state. However, philosophically, it is a completely different view of the notion of obligation – which can be confronted with agents' abilities as well.

19. Introducing epistemic properties and agents' abilities under uncertainty into DATL seems a natural extension of the scope of both ATOL and "Deontic ATL". Practical applications may include more realistic analysis of card games, security analysis, trust management as well as requirements engineering.

A similar list for hierarchies of models and multi-level decision making is shorter, but the problems it includes are not necessarily simpler:

1. In the simulations, all the agents were using very simple models of the reality, and very simple learning methods. The performance of more sophisticated hybrid agents should also be studied.

2. Using smarter (for instance, 2-level or 3-level) opponents may help to make the benefits of the proposed decision-making scheme more obvious.

3. Some experiments with human opponents can also be tried.

4. Using more complex models demands that the decision-making agent must possess more detailed knowledge as well. The hybrid agents had substantial problems with acquiring an accurate model of the reality. Thus, we may need a better exploration-exploitation scheme. Collaborative modeling (Zukerman and Albrecht, 2001) may also prove helpful, especially for models that include thousands or millions of states.

5. The experiments showed that the confidence measures used here were not perfect, hence this line of research can hardly be claimed completed.

Quoting old Bilbo – "the road goes on and on".[1]

[1] J.R.R. Tolkien, *Lord of the Rings*.

Appendix A

List of Acronyms

AETS	*alternating epistemic transition system*
ATEL	*alternating-time temporal epistemic logic*
ATEL-R*	*alternating-time temporal epistemic logic with recall*
ATL	*alternating-time temporal logic*
ATOL	*alternating-time temporal observational logic*
ATS	*alternating transition system*
BDI	*beliefs, desires and intentions* framework
BDI$_{CTL}$	the propositional modal logic of beliefs, desires and intentions with CTL as the temporal layer
CEM	*coalition effectivity model*
CGS	*concurrent game structure*
CL	*coalition logic* or *coalition game logic*
CTL	*computation tree logic*
DATL	*deontic* ATL
DTATL	*deontic* ATL *for temporal obligations*
ECL	*extended coalition logic*
MGM	*multi-player game model*
stit	the logic of *seeing to it that*.

Bibliography

Alchourron, C. (1993). Philosophical foundations of deontic logic and the logic of defeasible conditionals. In Meyer, J.-J. and Wieringa, R., editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 43–84. John Wiley & Sons.

Alur, R., de Alfaro, L., Henzinger, T. A., Krishnan, S., Mang, F., Qadeer, S., Rajamani, S., and Tasiran, S. (2000). MOCHA user manual. Technical report, University of Berkeley.

Alur, R., Henzinger, T. A., and Kupferman, O. (1997). Alternating-time Temporal Logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 100–109. IEEE Computer Society Press. Available at `http://citeseer.ist.psu.edu/170985.html`.

Alur, R., Henzinger, T. A., and Kupferman, O. (1998a). Alternating-time Temporal Logic. *Lecture Notes in Computer Science*, 1536:23–60. Available at `http://citeseer.ist.psu.edu/174802.html`.

Alur, R., Henzinger, T. A., and Kupferman, O. (2002). Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713. Updated, improved, and extended text. Available at `http://www.cis.upenn.edu/ alur/Jacm02.pdf`.

Alur, R., Henzinger, T. A., Kupferman, O., and Vardi, M. (1998b). Alternating refinement relations. In *CONCUR'98*.

Anderson, A. (1958). A reduction of deontic logic to alethic modal logic. *Mind*, 67:100–103.

Antoniou, G. (1999). A tutorial on default logics. *ACM Computing Surveys*, 31(4):337–359.

Aumann, R. and Hart, S., editors (1992). *Handbook of Game Theory with Economic Applications, Volume 1*. Elsevier/North-Holland.

Banerjee, B., Mukherjee, R., and Sen, S. (2000). Learning mutual trust. In *Working Notes of AGENTS-00 Workshop on Deception, Fraud and Trust in Agent Societies*, pages 9–14.

Belnap, N. (1991). Backwards and forwards in the modal logic of agency. *Philosophy and Phenomenological Research*, LI(4):777–807.

Belnap, N. and Perloff, M. (1988). Seeing to it that: a canonical form for agentives. *Theoria*, 54:175–199.

Berry, D. A. and Lindgren, B. W. (1996). *Statistics: Theory and Methods*. Wadsworth Publishing Company : Belmont, CA.

Bieber, P. and Cuppens, F. (1993). Expression of confidentiality policies with deontic logics. In Meyer, J.-J. and Wieringa, R., editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 103–123. John Wiley & Sons.

Billsus, D. and Pazzani, M. J. (1999). A hybrid user model for news story classification. In Kay, J., editor, *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 99–108. Springer.

Blum, A. L. and Furst, M. L. (1997). Fast planning through graph analysis. *Artificial Intelligence*, 90:281–300.

Bonnano, G. (2004). Memory requires von neumann games. *Synthese*. To appear in the section on *Knowledge, Rationality and Action*.

Broersen, J. (2003). *Modal Action Logics for Reasoning about Reactive Systems*. PhD thesis, Vrije Universiteit Amsterdam.

Broersen, J., Dastani, M., Huang, Z., and van der Torre, L. (2001a). The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In Müller, J., Andre, E., Sen, S., and Frasson, C., editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 9–16. ACM Press.

Broersen, J., Dastani, M., and van der Torre, L. (2001b). Resolving conflicts between beliefs, obligations, intentions, and desires. In Benferhat, S. and Besnard, P., editors, *Proceedings of ECSQARU 2001 (Symbolic and Quantitative Approaches to Reasoning with Uncertainty)*, pages 568–579.

Browne, W. J. and Draper, D. (2002). A comparison of Bayesian and likelihood-based methods for fitting multilevel models. Submitted. Available at `http://multilevel.ioe.ac.uk/team/materials/wbrssa.pdf`.

Bui, T. and Jamroga, W. (2003). Multi-agent planning with Planning Graph. In *Proceedings of eunite2003*, pages 558–565.

Carmel, D. and Markovitch, S. (1996). Learning and using opponent models in adversary search. Technical Report CIS9609, Technion, Haifa.

Chandra, A., Kozen, D., and Stockmeyer, L. (1981). Alternation. *Journal of the ACM*, 28(1):114–133.

Cohen, P. and Levesque, H. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42:213–261.

Corlett, R. and Todd, S. (1985). A Monte-Carlo approach to uncertain inference. In Ross, P., editor, *Proceedings of the Conference on Artificial Intelligence and Simulation of Behaviour*, pages 28–38.

Dignum, F. and Kuiper, R. (1997). Combining dynamic deontic logic and temporal logic for the specification of deadlines. In Sprague, J., editor, *Proceedings of HICSS*.

Doherty, P., Łukasiewicz, W., and Szałas, A. (2001). Computing strongest necessary and weakest sufficient conditions of first order formulas. In *Proceedings of IJCAI 2001*, pages 145–151.

Draper, D. (1995). Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society Series*, B(57):45–97.

Dunin-Kęplicz, B. and Verbrugge, R. (2003). A tuning machine for collective commitments. In Dunin-Kęplicz, B. and Verbrugge, R., editors, *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, pages 99–115.

Emerson, E. A. (1990). Temporal and modal logic. In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier Science Publishers.

Emerson, E. A. and Halpern, J. Y. (1986). "sometimes" and "not never" revisited: On branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178.

Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y. (1995). *Reasoning about Knowledge*. MIT Press: Cambridge, MA.

Fiadeiro, J. and Maibaum, T. (1991). Temporal reasoning over deontic specifications. *Journal of Logic and Computation*, 1(3):357–396.

Franceschet, M., Montanari, A., and de Rijke, M. (2000). Model checking for combined logics. In *Proceedings of the 3rd International Conference on Temporal Logic (ICTL)*.

Frank, I. (1996). *Search and Planning under Incomplete Information. A Study using Bridge Card Play*. PhD thesis, University of Edinburgh. Published in 1998 by Springer-Verlag.

Frank, I. and Basin, D. (1998). Search in games with incomplete information : A case study using bridge card play. *Artificial Intelligence*, 100(1-2):87–123.

Frank, I., Basin, D., and Matsubara, H. (1998). Finding optimal strategies for imperfect information games. In *AAAI/IAAI*, pages 500–507.

Gerbrandy, J. (1999). *Bisimulations on Planet Kripke*. PhD thesis, University of Amsterdam.

Ginsberg, M. (1996). How computers will play bridge. *The Bridge World*.

Ginsberg, M. (1999). GIB: Steps toward an expert-level bridge-playing program. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 584–589.

Ginsberg, M. (2001). GIB: Imperfect information in a computationally challenging game. *Journal of Articial Intelligence Research*.

Giunchiglia, F. and Traverso, P. (1999). Planning as model checking. In *ECP*, pages 1–20.

Goranko, V. (2001). Coalition games and alternating temporal logics. In van Benthem, J., editor, *Proc. of the 8th Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 259–272. Morgan Kaufmann. Corrected version. Available at `http://general.rau.ac.za/maths/goranko/papers/gltl31.pdf`.

Goranko, V. and Jamroga, W. (2004). Comparing semantics of logics for multi-agent systems. *Synthese*, 139(2):241–280. Series on on *Knowledge, Rationality and Action*.

Goranko, V. and Passy, S. (1992). Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2(1):5–30.

Goranko, V. and van Drimmelen, G. (2003). Complete axiomatization and decidability of the Alternating-time Temporal Logic. Submitted.

Hafer, T. and Thomas, W. (1987). Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In Baader, F., Brewka, G., and Eiter, T., editors, *Proceedings of 14th ICALP, Karlsruhe*, volume 267 of *Lecture Notes in Computer Science*, pages 269–279. Springer.

Hart, S. (1992). Games in extensive and strategic forms. In Aumann, R. and Hart, S., editors, *Handbook of Game Theory with Economic Applications, Volume 1*, pages 19–40. Elsevier/North-Holland.

Hintikka, J. and Sandu, G. (1997). Game-theoretical semantics. In van Benthem, J. and ter Meulen, A., editors, *Handbook of Logic and Language*, pages 361–410. Elsevier: Amsterdam.

Hoare, C. (1985). *Communicating Sequential Processes*. Prentice-Hall.

Hochreiter, S. and Mozer, M. C. (2001). Beyond maximum likelihood and density estimation: A sample-based criterion for unsupervised learning of complex models. *Advances in Neural Information Processing Systems*, 13. Proceedings of NIPS'2000.

Horty, J. and Belnap, N. (1995). The deliberative stit: A study of action, omission, ability, and obligation. *Journal of Philosophical Logic*, 24:583–644.

Huth, M. and Ryan, M. (2000). *Logic in Computer Science: Modeling and reasoning about systems*. Cambridge University Press.

Jamroga, W. (2001a). A defense model for games with incomplete information. In Baader, F., Brewka, G., and Eiter, T., editors, *KI 2001: Advances in Artificial Intelligence*, volume 2174 of *Lecture Notes in Artificial Intelligence*, pages 260–264. Springer. Proceedings of KI-2001.

Jamroga, W. (2001b). Multilevel modeling of dialogue environment for e-commerce agents. In Waldziński, D., editor, *Electronic Commerce - gospodarka XXI wieku*, pages 49–53.

Jamroga, W. (2002a). Datasize-based confidence measure for a learning agent. In Wiering, M., editor, *Benelearn 2002*, pages 39–45.

Jamroga, W. (2002b). Multiple models of reality and how to use them. In Blockeel, H. and Denecker, M., editors, *BNAIC 2002*, pages 155–162.

Jamroga, W. (2003a). Confidence measure for a learning agent. In et al., B. T., editor, *Proceedings of the Eighth Scandinavian Conference on Artificial Intelligence (SCAI'03)*, pages 95–106. IOS Press. Published in the series *Frontiers in Artificial Intelligence and Applications*.

Jamroga, W. (2003b). A confidence measure for learning probabilistic knowledge in a dynamic environment. In Mohammadian, M., editor, *Proceedings of International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA'03)*, pages 182–193.

Jamroga, W. (2003c). Safer decisions against a dynamic opponent. In M.A. Klopotek, S.T. Wierzchon, K. T., editor, *Intelligent Information Processing and Web Mining*, Advances in Soft Computing, pages 79–86. Springer. Proceedings of IIS 2003.

Jamroga, W. (2003d). Some remarks on Alternating Temporal Epistemic Logic. In Dunin-Kęplicz, B. and Verbrugge, R., editors, *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, pages 133–140.

Jamroga, W. (2004). Strategic planning through model checking of ATL formulae. To appear in the Proceedings of ICAISC'04, Lecture Notes in Computer Science.

Jamroga, W. and van der Hoek, W. (2003). Different semantics of Alternating-time Temporal Logic – a comparison. Unpublished.

Jamroga, W. and van der Hoek, W. (2004). Agents that know how to play. *Fundamenta Informaticae*. To appear.

Jamroga, W., van der Hoek, W., and Wooldridge, M. (2004). On obligations and abilities. In Lomuscio, A. and Nute, D., editors, *Deontic Logic in Computer Science: Proceedings of DEON 2004*, volume 3065 of *Lecture Notes in Computer Science*, pages 165–181. Springer Verlag.

Jonker, G. (2003). Feasible strategies in Alternating-time Temporal Epistemic Logic. Master thesis, University of Utrecht.

Kacprzak, M. and Penczek, W. (2004). Unbounded model checking for Alternating-time Temporal Logic. In *Proceedings of AAMAS-04*. To appear.

Kaelbling, L., Littman, M., and Moore, A. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Klir, G. J. (1999). Uncertainty and information measures for imprecise probabilities: An overview. In *Proceedings of the First International Symposium on Imprecise Probabilities and Their Applications*.

Klir, G. J. and Folger, T. (1988). *Fuzzy Sets, Uncertainty and Information*. Prentice Hall : Englewood Cliffs.

Kobsa, A. (1993). User modeling: Recent work, prospects and hazards. In Schneider-Hufschmidt, M., Kühme, T., and Malinowski, U., editors, *Adaptive User Interfaces: Principle and Practice*. Elsevier.

Koychev, I. (2000). Gradual forgetting for adaptation to concept drift. In *Proceedings of ECAI 2000 Workshop "Current Issues in Spatio-Temporal Reasoning"*, pages 101–106.

Koychev, I. (2001). Learning about user in the presence of hidden context. In *Proceedings of Machine Learning for User Modeling: UM-2001 Workshop*, pages 101–106. Available at `http://www.dfki.de/~rafer/um01-ml4um-ws/proceedings.html`.

Kumar, S. (1998). Confidence based dual reinforcement Q-routing: an on-line adaptive network routing algorithm. Master's thesis, Department of Computer Sciences, The University of Texas at Austin. Tech. Report AI98-267.

Kyburg, H. E. (1987). Bayesian and non-Bayesian evidential updating. *Artificial Intelligence*, 31:271–293.

Kyburg, H. E. (1988). Higher order probabilities and intervals. *International Journal of Approximate Reasoning*, 2:195–209.

Laroussinie, F. (1995). About the expressive power of CTL combinators. *Information Processing Letters*, 54(6):343–345.

Laroussinie, F. and Schnoebelen, P. (1995). A hierarchy of temporal logics with past. *Theoretical Computer Science*, 148(2):303–324.

Lei, Z., Saraydar, C. U., and Mandayam, N. B. (1999). Paging area optimization based on interval estimation in wireless personal communication networks. *Mobile Networks and Applications*, 5(1):85–99. Special Issue on Mobile Data Networks: Advanced Technologies and Services.

Levesque, H. (1990). All I know: a study in auto-epistemic logic. *Artificial Intelligence*, 42(3):263–309.

Lin, F. (2000). On strongest necessary and weakest sufficient conditions. In *Proceedings of KR 2000*, pages 167–175.

Lomuscio, A. and Sergot, M. (2002). On multiagent systems specification via deontic logic. In Meyer, J.-J. and Tambe, M., editors, *Intelligent Agents VIII : Proceedings of ATAL'01*, volume 2333 of *Lecture Notes in Computer Science*, pages 86–99. Springer.

Lomuscio, A. and Sergot, M. (2003a). Deontic interpreted systems. *Studia Logica*, 75(1):63–92.

Lomuscio, A. and Sergot, M. (2003b). A formalization of violation, error recovery, and enforcement in the bit transmission problem. *Journal of Applied Logic*, 1. To appear. Preliminary version published in the Proceedings of DEON'02.

Macieszczak, A. and Mikke, J. (1987). *Bez impasu [No Finesse]*. Mlodziezowa Agencja Wydawnicza : Warsaw.

Maibaum, T. (1993). Temporal reasoning over deontic specifications. In Meyer, J.-J. and Wieringa, R., editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 141–202. John Wiley & Sons.

Mally, E. (1926). *Grundgesetze des Sollens, Elemente der Logik des Willens*. Leuschner & Lubensky.

Marshall, E. C. and Spiegelhalter, D. J. (1999). Strategies for inference robustness in complex modelling: An application to longitudinal performance measures. Technical report, MRC Biostatistics Unit, Cambridge, UK.

Mengusoglu, E. and Ris, C. (2001). Use of acoustic prior information for confidence measure in ASR applications. In *Eurospeech 2001*.

Merhav, N. and Feder, M. (1998). Universal prediction. *IEEE Trans. Inform. Theory*, IT-44(6):2124–2147.

Meyer, J.-J. (1988). A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic*, 29(1):109–136.

Meyer, J.-J. and Wieringa, R. (1993a). Deontic logic: A concise overview. In Meyer, J.-J. and Wieringa, R., editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 3–16. John Wiley & Sons.

Meyer, J.-J. and Wieringa, R., editors (1993b). *Deontic Logic in Computer Science: Normative System Specification*. John Wiley & Sons.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Moore, R. (1985). A formal theory of knowledge and action. In Hobbs, J. and Moore, R., editors, *Formal Theories of the Commonsense World*. Ablex Publishing Corp.

Morgenstern, L. (1991). Knowledge and the frame problem. *International Journal of Expert Systems*, 3(4).

Moses, Y. and Tennenholz, M. (1990). Artificial social systems part I: Basic principles. Technical Report CS90-12, Weizmann Institute.

Moses, Y. and Tennenholz, M. (1995). Artificial social systems. *Computers and AI*, 14(6):533–562.

Myerson, R. (1999). Nash equilibrium and the history of economic theory. *Journal of Economic Literature*, 37:1067–1082.

Nash, J. (1950). Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences U.S.A.*, 36:48–49.

Osborne, M. and Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press.

Parikh, R. (1985). The logic of games and its applications. *Ann. of Discrete Mathematics*, 24:111–140.

Pauly, M. (2000). Game logic for game theorists. Technical Report INS-R0017, CWI.

Pauly, M. (2001a). *Logic for Social Software*. PhD thesis, University of Amsterdam.

Pauly, M. (2001b). A logical framework for coalitional effectivity in dynamic procedures. *Bulletin of Economic Research*, 53(4):305–324.

Pauly, M. (2002). A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166.

Pearl, J. (1987). Do we need higher-order probabilities and, if so, what do they mean? In *Uncertainty in Artificial Intelligence Workshop*.

Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*. MIT Press : Cambridge, Mass.

Pistore, M. and Traverso, P. (2001). Planning as model checking for extended goals in non-deterministic domains. In *Proceedings of IJCAI*, pages 479–486.

Quine, W. (1956). Quantifiers and propositional attitudes. *Journal of Philosophy*, 53:177–187.

Rao, A. and Georgeff, M. (1991). Modeling rational agents within a BDI-architecture. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484.

Rao, A. and Georgeff, M. (1995). Formal models and decision procedures for multi-agent systems. Technical Report 61, Australian Artificial Intelligence Institute.

Russel, S. and Norvig, P. (1995). *Artificial Intelligence: a Modern Approach*. Prentice Hall.

Schild, K. (2000). On the relationship between BDI logics and standard logics of concurrency. *Autonomous Agents and Multi Agent Systems*, pages 259–283.

Schnoebelen, P. (2003). The complexity of temporal model checking. In *Advances in Modal Logics, Proceedings of AiML 2002*. World Scientific.

Schobbens, P. Y. (2003). ATL with imperfect recall. Under preparation. Presented at the workshop "Logic and Communication in Multi-Agent Systems", June 29, 2003, Eindhoven, Netherlands.

Sen, S. and Arora, N. (1997). Learning to take risks. In *Working Papers of the AAAI-97 Workshop on Multiagent Learning*, pages 59–64.

Sen, S. and Sekaran, M. (1998). Individual learning of coordination knowledge. *Journal of Experimental & Theoretical Artificial Intelligence*, 10:333–356.

Sen, S. and Weiss, G. (1999). Learning in multiagent systems. In G., W., editor, *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, pages 259–298. MIT Press: Cambridge, Mass.

Shoham, Y. and Tennenholz, M. (1992). On the synthesis of useful social laws for artificial agent societies. In *Proceedings of AAAI-92*.

Shoham, Y. and Tennenholz, M. (1997). On the emergence of social conventions: Modelling, analysis, and simulations. *Artificial Intelligence*, 94(1-2):139–166.

Shubik, M. (1998). Game theory, complexity, and simplicity. Part III: Critique and prospective. Technical Report 98-04-029, Santa Fe Institute.

Spiegelhalter, D. J., Best, N. G., and Carlin, B. P. (1998). Bayesian deviance, the effective number of parameters, and the comparison of arbitrarily complex models. Technical report, MRC Biostatistics Unit, Cambridge, UK.

van Benthem, J. (2000). Logic and games. Technical Report X-2000-03, ILLC. Lecture notes, available at `http://www.illc.uva.nl/Publications/Lists/www-allpubl-2000.html`.

van Benthem, J. (2001). Games in dynamic epistemic logic. *Bulletin of Economic Research*, 53(4):219–248. Proceedings of LOFT-4.

van Benthem, J. (2002). The epistemic logic of IF games (hintikka self-applied). Under preparation. To appear in the Hintikka Volume, *Library of Living Philosophers*. Available at `http://staff.science.uva.nl/~johan/H-H.pdf`.

van der Hoek, W., Roberts, M., and Wooldridge, M. (2004). Social laws in alternating time: Effectiveness, feasibility and synthesis. Submitted.

van der Hoek, W. and Verbrugge, R. (2002). Epistemic logic: A survey. *Game Theory and Applications*, 8:53–94.

van der Hoek, W. and Wooldridge, M. (2002). Tractable multiagent planning for epistemic goals. In Castelfranchi, C. and Johnson, W., editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-02)*, pages 1167–1174. ACM Press, New York.

van der Hoek, W. and Wooldridge, M. (2003a). Cooperation, knowledge and time: Alternating-time Temporal Epistemic Logic and its applications. *Studia Logica*, 75(1):125–157.

van der Hoek, W. and Wooldridge, M. (2003b). Model checking cooperation, knowledge and time – a case study. *Research in Economics*, 57(3):235–265.

van der Hoek, W. and Wooldridge, M. (2003c). Towards a logic of rational agency. *Logic Journal of the IGPL*, 11(2):135–160.

van Ditmarsch, H. (2000). *Knowledge Games*. PhD thesis, University of Groningen.

van Drimmelen, G. (2003). Satisfiability in Alternating-time Temporal Logic. In *Proceedings of LICS'2003*, pages 208–217. IEEE Computer Society Press.

van Eck, J. (1982). A system of temporally relative modal and deontic predicate logic and its philosophical applications. *Logique et Analyse*, 100:249–381.

van Linder, B., van der Hoek, W., and Meyer, J.-J. C. (1998). Formalizing abilities and opportunities of agents. *Fundamenta Informaticae*, 34(1,2):53–101.

van Otterloo, S. (2004). Preferences in game logics. In *Proceedings of AAMAS-04*. To appear.

van Otterloo, S. and Jonker, G. (2004). On Epistemic Temporal Strategic Logic. Submitted.

van Otterloo, S., van der Hoek, W., and Wooldridge, M. (2003). Knowledge as strategic ability. *Electronic Lecture Notes in Theoretical Computer Science*, 85(2).

Vidal, J. and Durfee, E. (1998). Learning nested agent models in an information economy. *Journal of Experimental and Theoretical Artificial Intelligence*, 10:291–308.

von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behaviour*. Princeton University Press: Princeton, NJ.

von Wright, G. (1951). Deontic logic. *Mind*, 60:1–15.

von Wright, G. (1964). A new system of deontic logic. *Danish Yearbook of Philosophy*, 1:173–182.

Wang, P. (2001). Confidence as higher-order uncertainty. In *Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications*, pages 352–361.

Wang, R., Zhang, H.-J., and Zhang, Y.-Q. (1999). A confidence measure based moving object extraction system built for compressed domain. Technical report, Microsoft Research, Beijing.

Weiss, G., editor (1999). *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press: Cambridge, Mass.

Widmer, G. (1997). Tracking context changes through meta-learning. *Machine Learning*, 27:256–286.

Wieringa, R. and Meyer, J.-J. (1993). Applications of deontic logic in computer science: A concise overview. In Meyer, J.-J. and Wieringa, R., editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 17–40. John Wiley & Sons.

Wieringa, R., Meyer, J.-J., and Weigand, H. (1989). Specifying dynamic and deontic integrity constraints. *Data and Knowledge Engineering*, 4:157–189.

Williams, G. (1999). *Knowing What You Don't Know: Roles for Confidence Measures in Automatic Speech Recognition*. PhD thesis, University of Sheffield, Department of Computer Science.

Williams, G. and Renals, S. (1997). Confidence measures for hybrid HMM/ANN speech recognition. In *Eurospeech97*, pages 1955–1958.

Winograd, T. and Flores, F. (1986). *Understanding Computers and Cognition*. Ablex Publishing.

Wooldridge, M. (2000). *Reasoning about Rational Agents*. MIT Press : Cambridge, Mass.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley & Sons.

Zukerman, I. and Albrecht, D. W. (2001). Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11:5–18.

Zukerman, I., Albrecht, D. W., and Nicholson, A. (1999). Predicting users' requests on the WWW. In *UM99 – Proceedings of the Seventh International Conference on User Modeling*.

Summary

This thesis considers some aspects of multi-agent systems, seen as a metaphor for reasoning about the world, and providing a conceptual machinery that can be used to model and analyze the reality in which an agent is embedded. First, we study several modal logics for multi-agent systems; in particular, Alternating-time Temporal Logic (ATL) is studied in various contexts. Then, a concept of multi-level modeling of reality and multi-level decision making is proposed in the second part of the thesis.

The formal investigation of ATL yields equivalence results for several different semantics of this language, as well as a thorough comparison of ATL and Coalition Logic – another recent logic of strategic ability. We also study an epistemic extension of ATL, show a satisfiability preserving interpretation of the extension into the "pure" ATL, and demonstrate its similarities to the well known BDI logic of beliefs, desires and intentions. After that, we point out some counterintuitive features of this particular extension, and propose how it can be recovered. The extension can be also seen as a generic scheme of enriching game theory-based logics with other concepts and dimensions. To support this, we propose how ATL can be extended with the deontic notion of obligation. Apart of investigating cooperation modalities and their semantics as the formal counterpart of the decision making process, we devise efficient procedures for multi-agent planning to achieve goals that can be expressed with formulae of ATL and some of its extensions.

In the second part of the thesis, a concept of multi-level modeling of reality is proposed, where various models of the same environment can be combined to improve decision making. Our main motivation is to endow agents with some way of using secure game theory-based models together with adaptive models obtained through machine learning. We try several confidence measures, and verify the whole idea through experiments with simple software agents in an e-banking scenario.

Samenvatting

Dit proefschrift behandelt een aantal aspecten van muti-agent systemen, welke gezien kunnen worden als een metafoor voor redeneren over de wereld, en bieden tevens conceptuele gereedschappen aan die gebruikt kunnen worden om de realiteit waarin de agent zich bevindt te modeleren en analyseren.

Als eerste worden verschillende modale logica's voor multi agent systemen bekeken, in het bijzonder wordt de Alternating-time Temporal Logic (ATL) in verschillende contexten bestudeerd.

Vervolgens wordt een concept van multi-level modellering van de realiteit en multi-level decision making voorgesteld in het tweede deel van het proefschrift.

Het formele onderzoek van ATL leidt tot resultaten van equivalentie voor verscheidene semantieken van deze taal alsmede een grondige vergelijking van ATL en Coalition Logic, een andere recente logica van strategische mogelijkheden.

We bestuderen ook een epistemische uitbreiding van ATL, laten een satisfiability behoudende interpretatie van de uitbreiding in de "pure" ATL zien en demonstreren de gelijkenis met de bekende BDI logica van geloof, verlangens en intenties.

Hierna stippen we enige tegenintuitieve kenmerken van deze specifieke uitbreiding aan en doen een voorstel hoe deze kunnen worden opgelost. De uitbreiding kan ook gezien worden als een generiek schema voor verrijking van op speltheorie gebaseerde logica's met andere concepten en dimensies.

Om dit te ondersteunen doen we een voorstel hoe ATL uitgebreid kan worden met de deontische notie van verplichting. Naast het onderzoek naar de samenwerking van modaliteiten en hun semantiek als formele tegenhanger van het besluitvormingsproces, laten we efficiente procedures zien die doelgerichte multi-agent planning uitdrukken geformuleerd in ATL en enkele uitbreidingen daarvan.

In het tweede deel van de thesis wordt een concept van multi-level modellering van de werkelijkheid voorgesteld, waarmee verschillende modellen van de werkelijkheid kunnen worden gecombineerd om zo het besluitvormings proces te verbeteren. Onze belangrijkste motivatie is om agents uit te rusten met een manier om veilige speltheoretische modellen en middels machine learning verkregen adaptieve modellen naast elkaar te kunnen gebruiken.

Verschillende betrouwbaarheidsmaten worden uitgetest en het geheel wordt geverifieerd met behulpvan experimenten met eenvoudige software agents in een e-banking scenario.

Streszczenie

Tematem niniejszej pracy są niektóre aspekty systemów wieloagentowych, widzianych jako sposób modelowania świata i rozumowania na temat otaczającej rzeczywistości. Na pierwszą część pracy składa się studium wybranych logik modalnych do opisu środowisk agentowych – a zwłaszcza tak zwanej logiki temporalnej czasu alternującego (Alternating-time Temporal Logic, ATL). Druga część pracy przedstawia koncepcję wielopoziomowego modelowania rzeczywistości i wielopoziomowego podejmowania decyzji.

Badania nad ATL wykazują równoważność kilku alternatywnych semantyk tego języka. Coalition Logic, logika pokrewna ATL – i również inspirowana teorią gier – okazuje się operować tym samym aparatem konceptualnym (lecz mniejszą siłą wyrazu). Istotnym elementem pierwszej części pracy są studia nad epistemicznym rozszerzeniem ATL, zwanym ATEL: demonstrujemy jak interpretowac formuly i modele ATEL w "czystym" ATL (z zachowaniem spełnialności i prawdziwości formuł w modelach), a także porównujemy ATEL ze znanym formalizmem BDI (*beliefs, desires and intentions*: przekonania, pragnienia i intencje). Następnie wskazujemy pewne aspekty ATEL, które wydają się być sprzeczne z intuicją, i proponujemy dwa alternatywne sposoby "naprawy" tej logiki.

Logiki typu ATL można w podobny sposób rozszerzać także o inne aspekty agentów i ich interakcji. Jako przykład posłużyć może zaproponowany w niniejszej pracy "Deontyczny ATL", rozszerzający oryginalną logikę o deontyczne pojęcie zobowiązania. Jako że modalności kooperacyjne (leżące u podstawy ATL) stanowią de facto formalny odpowiednik procesu podejmowania decyzji, niniejsza praca prezentuje też efektywne algorytmy planowania w środowiskach wieloagentowych dla celów dających się wyrazić formułami ATL i niektorych jej rozszerzeń.

Przedmiotem drugiej części pracy jest koncepcja wielopoziomowego modelowania rzeczywistości: jeśli agent posiada kilka alternatywych modeli rzeczywistości, w niektórych sytuacjach może on uzyskać lepsze rezultaty, używając kombinacji wszystkich tych modeli, niż wybierając tylko jeden z nich i rezygnując z pozostałych. Taki schemat podejmowania decyzji może, między innymi, posłużyć agentom do znaleziena rozsądnej równowagi pomiędzy bezpiecznymi rozwiązaniami opartymi o teorię gier, a adaptywnymi modelami świata uzyskanymi poprzez maszynowe uczenie. Poszukując takiego punktu równowagi, wypróbowujemy kilka alternatywnych miar zaufania, by w końcu zweryfikować przedstawianą ideę przy użyciu symulacji prostej gry opartej o scenariusz bankowości elektronicznej.

SIKS Dissertation Series

1998-1 .... Johan van den Akker (CWI), *DEGAS - An Active, Temporal Database of Autonomous Objects*

1998-2 .... Floris Wiesman (UM), *Information Retrieval by Graphically Browsing Meta-Information*

1998-3 .... Ans Steuten (TUD), *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*

1998-4 .... Dennis Breuker (UM), *Memory versus Search in Games*

1998-5 .... E.W.Oskamp (RUL), *Computerondersteuning bij Straftoemeting*

1999-1 .... Mark Sloof (VU), *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*

1999-2 .... Rob Potharst (EUR), *Classification using decision trees and neural nets*

1999-3 .... Don Beal (UM), *The Nature of Minimax Search*

1999-4 .... Jacques Penders (UM), *The practical Art of Moving Physical Objects*

1999-5 .... Aldo de Moor (KUB), *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*

1999-6 .... Niek J.E. Wijngaards (VU), *Re-design of compositional systems*

1999-7 .... David Spelt (UT), *Verification support for object database design*

1999-8 .... Jacques H.J. Lenting (UM), *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.*

2000-1 .... Frank Niessink (VU), *Perspectives on Improving Software Maintenance*

2000-2 . . . . Koen Holtman (TUE), *Prototyping of CMS Storage Management*

2000-3 . . . . Carolien M.T. Metselaar (UVA), *Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.*

2000-4 . . . . Geert de Haan (VU), *ETAG, A Formal Model of Competence Knowledge for User Interface Design*

2000-5 . . . . Ruud van der Pol (UM), *Knowledge-based Query Formulation in Information Retrieval.*

2000-6 . . . . Rogier van Eijk (UU), *Programming Languages for Agent Communication*

2000-7 . . . . Niels Peek (UU), *Decision-theoretic Planning of Clinical Patient Management*

2000-8 . . . . Veerle Coupé (EUR), *Sensitivity Analyis of Decision-Theoretic Networks*

2000-9 . . . . Florian Waas (CWI), *Principles of Probabilistic Query Optimization*

2000-10 . . . . Niels Nes (CWI), *Image Database Management System Design Considerations, Algorithms and Architecture*

2000-11 . . . . Jonas Karlsson (CWI), *Scalable Distributed Data Structures for Database Management*

2001-1 . . . . Silja Renooij (UU), *Qualitative Approaches to Quantifying Probabilistic Networks*

2001-2 . . . . Koen Hindriks (UU), *Agent Programming Languages: Programming with Mental Models*

2001-3 . . . . Maarten van Someren (UvA), *Learning as problem solving*

2001-4 . . . . Evgueni Smirnov (UM), *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*

2001-5 . . . . Jacco van Ossenbruggen (VU), *Processing Structured Hypermedia: A Matter of Style*

2001-6 . . . . Martijn van Welie (VU), *Task-based User Interface Design*

2001-7 . . . . Bastiaan Schonhage (VU), *Diva: Architectural Perspectives on Information Visualization*

2001-8 . . . . Pascal van Eck (VU), *A Compositional Semantic Structure for Multi-Agent Systems Dynamics.*

2001-9 . . . . Pieter Jan 't Hoen (RUL), *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*

2001-10 . . . . Maarten Sierhuis (UvA), *Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design*

2001-11 . . . . Tom M. van Engers (VUA), *Knowledge Management: The Role of Mental Models in Business Systems Design*

2002-01 . . . . Nico Lassing (VU), *Architecture-Level Modifiability Analysis*

2002-02 . . . . Roelof van Zwol (UT), *Modelling and searching web-based document collections*

2002-03 . . . . Henk Ernst Blok (UT), *Database Optimization Aspects for Information Retrieval*

2002-04 . . . . Juan Roberto Castelo Valdueza (UU), *The Discrete Acyclic Digraph Markov Model in Data Mining*

2002-05 . . . . Radu Serban (VU), *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*

2002-06 . . . . Laurens Mommers (UL), *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*

2002-07 . . . . Peter Boncz (CWI), *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*

2002-08 . . . . Jaap Gordijn (VU), *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*

2002-09 . . . . Willem-Jan van den Heuvel(KUB), *Integrating Modern Business Applications with Objectified Legacy Systems*

2002-10 . . . . Brian Sheppard (UM), *Towards Perfect Play of Scrabble*

2002-11 . . . . Wouter C.A. Wijngaards (VU), *Agent Based Modelling of Dynamics: Biological and Organisational Applications*

2002-12 . . . . Albrecht Schmidt (Uva), *Processing XML in Database Systems*

2002-13 . . . . Hongjing Wu (TUE), *A Reference Architecture for Adaptive Hypermedia Applications*

2002-14 . . . . Wieke de Vries (UU), *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*

2002-15 . . . . Rik Eshuis (UT), *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*

2002-16 . . . . Pieter van Langen (VU), *The Anatomy of Design: Foundations, Models and Applications*

2002-17 .... Stefan Manegold (UVA), *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003-01 .... Heiner Stuckenschmidt (VU), *Ontology-Based Information Sharing in Weakly Structured Environments*

2003-02 .... Jan Broersen (VU), *Modal Action Logics for Reasoning About Reactive Systems*

2003-03 .... Martijn Schuemie (TUD), *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*

2003-04 .... Milan Petkovic (UT), *Content-Based Video Retrieval Supported by Database Technology*

2003-05 .... Jos Lehmann (UVA), *Causation in Artificial Intelligence and Law - A modelling approach*

2003-06 .... Boris van Schooten (UT), *Development and specification of virtual environments*

2003-07 .... Machiel Jansen (UvA), *Formal Explorations of Knowledge Intensive Tasks*

2003-08 .... Yongping Ran (UM), *Repair Based Scheduling*

2003-09 .... Rens Kortmann (UM), *The resolution of visually guided behaviour*

2003-10 .... Andreas Lincke (UvT), *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*

2003-11 .... Simon Keizer (UT), *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*

2003-12 .... Roeland Ordelman (UT), *Dutch speech recognition in multimedia information retrieval*

2003-13 .... Jeroen Donkers (UM), *Nosce Hostem - Searching with Opponent Models*

2003-14 .... Stijn Hoppenbrouwers (KUN), *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*

2003-15 .... Mathijs de Weerdt (TUD), *Plan Merging in Multi-Agent Systems*

2003-16 .... Menzo Windhouwer (CWI), *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses*

2003-17 .... David Jansen (UT), *Extensions of Statecharts with Probability, Time, and Stochastic Timing*

2003-18 .... Levente Kocsis (UM), *Learning Search Decisions*

2004-01 .... Virginia Dignum (UU), *A Model for Organizational Interaction: Based on Agents, Founded in Logic*

2004-02 .... Lai Xu (UvT), *Monitoring Multi-party Contracts for E-business*

2004-03 .... Perry Groot (VU), *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*

2004-04 .... Chris van Aart (UVA), *Organizational Principles for Multi-Agent Architectures*

2004-05 .... Viara Popova (EUR), *Knowledge discovery and monotonicity*

2004-06 .... Bart-Jan Hommes (TUD), *The Evaluation of Business Process Modeling Techniques*

2004-07 .... Elise Boltjes (UM), *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*

2004-08 .... Joop Verbeek(UM)), *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politile gegevensuitwisseling en digitale expertise*

2004-09 .... Martin Caminada (VU), *For the Sake of the Argument; explorations into argument-based reasoning*

2004-10 .... Suzanne Kabel (UVA), *Knowledge-rich indexing of learning-objects*

2004-11 .... Michel Klein (VU), *Change Management for Distributed Ontologies*

2004-12 .... The Duy Bui (UT), *Creating emotions and facial expressions for embodied agents*

2004-13 .... Wojciech Jamroga (UT), *Using Multiple Models of Reality: On Agents who Know how to Play Safer*