A LITTLE ABOUT ACTION, KNOWLEDGE, BELIEF AND A LOT ABOUT MODAL LOGIC

Tiago de Lima

Centre de Recherche en Informatique de Lens
UFR de Sciences Jean Perrin
Rue Jean Souvraz SP 18
62307 Lens Cedex
France

phone: +33 3 21 79 17 23
email: gestion@cril.univ-artois.fr
homepage: http://www.cril.univ-artois.fr/

A LITTLE ABOUT ACTION, KNOWLEDGE, BELIEF AND A LOT ABOUT MODAL LOGIC

Tiago de Lima

Submitted in partial fulfilment
of the requirements for the
"HABILITATION À DIRIGER DES RECHERCHES"

Examining committee:

Referees:

Anthony Hunter	Full professor at University College London, UK
Odile Papini	Full professor at Aix-Marseille University, France
Umberto Straccia	Senior researcher at CNR, Italy

Members:

Salem Benferhat	Full professor at Artois University, France
Philippe Besnard	Senior researcher at CNRS, France
Sébastien Konieczny	Senior researcher at CNRS, France

ARTOIS UNIVERSITY
December 5, 2019

Acknowledgements

This is fruit of almost eleven years of research work. I would like to list here the names of all people that contributed to this achievement. However, I am afraid I will not be able to remember everyone. Therefore, if you fell that your name is missing from the list below, please, accept my apologies and also my sincere acknowledgements right now.

First of all, I would like to thank Salem Benferhat for the advice and friendship. Thanks to the referees Anthony Hunter, Odile Papini and Umberto Straccia, as well as the members of the examining committee Philippe Besnard, and Sébastien Konieczny.

Thanks to all co-authors of papers during this period Philippe Balbiani, Thomas Caridroit, Frank Dignum, Hans van Ditmarsch, Andreas Herzig, Sébastien Konieczny, Jean-Marie Lagniez, Sébastien Magnier, Pierre Marquis, Valentin Montmirail, Daniel Le Berre, Emiliano Lorini, Lambèr Royakkers, and Nicolas Troquard without whom this work would never have existed.

Special thanks to Lorena Beghetto Marli de Lima, Antonio de Lima, and Mousse.

Thanks to the Artois University, the Centre de Recherche en Informatique de Lens and the Eindhoven University of Technology where all the work has been done.

And finally, thank you, the reader, for reading this thesis.

Tiago de Lima
Lille, October 2019

Contents

Chapter 1

Introduction

As the reader may have already guessed, this work is all about modal logics for actions, knowledge and belief, but also some additional concepts, such as responsibility and ability. Indeed, this is the kind of research I decided to do since my Ph.D., more than a decade ago. This topic is part of a broader subarea of artificial intelligence called knowledge representation and reasoning. The study described here intersects some subareas of philosophy and also multi-agent systems, game theory and computational complexity.

This work synthesises research results obtained after my Ph.D., from 2008 onward. Starting on Chapter 4, the structure of the document is organised in order to reflect some of the main topics I worked on during this period. I chose the subjects that are more representative of my work.

But, before that, Chapter 2 presents, as its title suggests, some formal preliminaries. The idea is to establish the mathematical notation that will be used in the entire document, as well as to recall basic definitions, theorems and facts that will be referred to in the subsequent chapters. This exposition is divided into two parts. Section 2.1 presents classical propositional logic (hereafter CPL), its syntax, semantics, axiom systems and reasoning methods. Section 2.2 contains a very brief introduction to belief revision theory. The AGM postulates and a couple of fundamental theorems are presented in order to pave the way for Chapter 6, that will refer to that material. After that, Chapter 3 presents modal logics. We will see modal logic syntax, semantics, axiom systems, expressiveness, reasoning methods and also some applications.

Chapter 4 presents a formalism aiming at modeling responsibility in multi-agents environments. I started working on this subject during my two years post-doc at the Eindhoven University of Technology, in The Netherlands. There, I wrote some papers on logic and responsibility with Lambèr Royakkers from Eindhoven and Frank Dignum from the Utrecht University. The most recent work I published on that topic was done some years latter, after I moved to France.

The formalism is called coalition epistemic dynamic logic (CEDL). It is a modal

logic that contains epistemic and dynamic operators. Other operators are defined as abbreviations, such as obligations, ability and knowing ability. For instance, an agent has the ability to achieve φ if there is an action that the agent can execute and its execution leads to φ. But, knowing ability means that the agent, in addition, knows what action it is. A sound and complete axiom system is also provided. On the conceptual side, CEDL has been designed to reason about responsibility, which is a concept defined in terms of causality, knowledge and obligation. We will see formal definitions of different kinds of individual and collective responsibility and, consequently, some logical relations between these different concepts.

Chapter 5 presents a formalism aiming at modeling agents abilities and knowledge trough time. That work was carried out at the Lens Computer Science Research Laboratory (abbreviated CRIL), in France. I joined the CRIL mid 2009 to occupy a position of Lecturer (*maître de conférences*). The position was offered along with a very interesting five years CNRS Chair (*chaire CNRS*).

The logic we see in Chapter 5 is called alternating-time temporal dynamic epistemic logic (ATDEL). ATDEL also has dynamic and knowledge operators and, in addition, temporal operators. The dynamic operators of ATDEL are designed in "dynamic epistemic logic style". Here, we only define public actions, such as public announcements and also factual change actions that are perceived by all agents in the environment. In that respect, ATDEL is less expressive than the former CEDL. However, ATDEL semantics permits much shorter system specifications. In addition, apart from a sound and complete axiom system, algorithms for model checking and satisfiability checking in an interesting fragment of the logic are proposed. We will also see a comparison of the expressiveness of ATDEL with several different logics from the literature.

Chapter 6 presents some research on multi-agent belief change. This work has been carried out in collaboration with CRIL colleagues Thomas Caridroit, Sébastien Konieczny and Pierre Marquis. Thomas was my first Ph.D. student. I co-supervised him together with Sébastien and Pierre from 2013 until 2016. Working with Thomas, Sébastien and Pierre on this subject, which has a substantial intersection with, but is slightly different from, what I have been done before, was rewarding. It also permitted the opening of new perspectives for my research.

CEDL and ATDEL were designed with knowledge operators. One may think that these logics could work well with belief, and the change would just be a small technical trick. But this is not the case. I can tell, I tried it. When belief operators naively replace knowledge in the formalisms mentioned above, the resultant logic just does not work as it should.

When an agent faces a new piece of information that contradicts previous beliefs, a revision must be done. This can be a quite complicated process. "Correct" ways of revising beliefs have been proposed in the field of belief revision theory. Chapter 6 presents an adaptation of these techniques to multi-agents scenarios. AGM belief expansion and revision postulates are generalised to such scenarios and concrete operators for multi-agent expansion and revision are proposed.

Chapter 7 presents some research on automated reasoning for modal logics. This time, the study was performed in collaboration with my CRIL colleagues Jean-Marie Lagniez, Daniel Le Berre and Valentin Montmirail. Valentin was my second Ph.D.

student. I co-supervised him with Jean-Marie and Daniel. Once again, the experience was rewarding, and even more perspectives have been opened. I consider this topic particularly important for my personal objectives. The possibility to develop practical applications from what was, up to some years ago, exclusively theoretical research seems promising.

On that chapter, we will see methods for satisfiability checking in modal logics K and KT5 (also known as S5). The method for KT5 amounts to an efficient translation from that logic to classical propositional logic that is handed over to a performant SAT solver. Practical experiments show that this technique outperformed all alternative methods. The reasoning method for modal logic K also uses a translation to CPL. But, since the translated formula can be exponentially larger than the original one, a more clever algorithm was designed in order to work with "parts" of that translation.

Finally, Chapter 8 concludes the thesis. It discusses some results and points out new possible directions of research.

This document does not contain all the research I have being doing since 2008. For instance, still in Eindhoven, I published a couple of papers on acceptance logic and intentions and plan dynamics with Andreas Herzig and Emiliano Lorini both from Toulouse. After moving to France, some new papers on dynamic epistemic logics, co-authored with Philippe Balbiani, Hans van Ditmarsch and Andreas Herzig have been published as well. There is also a fine paper on dialogical logic written with Sébastien Magnier, at the time, a Ph.D. student supervised by Shahid Rahman from Lille. Appendix A contains a list of all my publications on the period.

The great majority of proofs of theorems and alike are omitted from the this text. Only some "easy" proofs and sketches are present, and only when they help understanding the results. This decision was motivated by the fact that long or difficult proofs may disrupt the reading process, take away the focus or, worse, annoy the reader. In each case, the reader will be pointed to the articles where these proofs can be found.

Chapter 2

Formal Preliminaries

2.1 Classical Propositional Logic

In the very beginning of his class notes, Fitting (2010) says that:

> "Classical propositional logic is the simplest and most nicely behaved of any logic (whatever that means)."

Since I do not know what that means, I prefer to motivate the study of *classical propositional logic (CPL)* (also called sentential logic, statement logic, propositional calculus, etc.) by the fact that it is one of the most fundamental logical formalisms we have. It is included in modal logics and thus also in first-order logic and higher-order logic.

CPL studies *propositions*, which are sentences to which one can assign truth values, i.e., they can be true or false. For example, the following two sentences are propositions:

(2.1) John teaches at the university

(2.2) John is an academic.

In addition to propositions, CPL also has connectives, such as $\rightarrow$ (material implication) and $\wedge$ (conjunction), that can be used to form new propositions. For instance, the sentence:

(2.3) If John teaches at the university then John is an academic.

is also a proposition. This proposition is true if and only if, if (2.1) is true then (2.2) is also true. If we are able to show that both (2.1) and (2.3) are true, CPL permits us to conclude that (2.2) is true as well.

On the next subsections, we see the syntax and semantics of CPL, which permit us to construct propositions and to give them meaning. After that, we see different proof methods for CPL, which is what permits us to study the relationships among propositions and to draw conclusions from them.

2.1.1 Syntax

The symbols used in the language of CPL are *constants*, *propositional variables* (also called propositional letters) and *connectives*. We assume the constant $\bot$ and a non-empty countable (possibly infinite) set of propositional variables $\mathbb{P}$. Elements of $\mathbb{P}$ are noted p_0, p_1, p_2, etc.. The *primitive connectives* used here are $\rightarrow$, $\wedge$ and $\vee$. The first one is called 'material implication' (or 'if-then', if you prefer), the other two are called, respectively, 'conjunction' and 'disjunction'. Some other (non-primitive) connectives are also used. They are defined as abbreviations. Formally, we have the following.

Definition 2.1 (CPL Formula)**.** Let $\mathbb{P}$ be a non-empty countable (possibly infinite) set of propositional variables. The *language of classical propositional logic*, also called the set of *classical propositional logic formulas*, is noted $\mathcal{L}_{\text{CPL}}$ and is defined by the following grammar in Bakus-Nahur form:

$$\varphi ::= \bot \mid p \mid (\varphi \rightarrow \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi)$$

where p is any element of $\mathbb{P}$.

Throughout this document, we also use the constant $\top$ and connectives $\neg$ and $\leftrightarrow$. They are defined as abbreviations, namely: $\top \stackrel{\text{def}}{=} \neg\bot$, $\neg\varphi \stackrel{\text{def}}{=} (\varphi \rightarrow \bot)$ and $(\varphi \leftrightarrow \psi) \stackrel{\text{def}}{=} ((\varphi \rightarrow \psi) \wedge (\varphi \rightarrow \psi))$. We sometimes omit outer parentheses when convenient.

A formula of the form $(\varphi \rightarrow \psi)$ is read 'if φ then ψ' (or, 'φ implies ψ'), a formula of the form $(\varphi \wedge \psi)$ is read 'φ and ψ' and a formula of the form $(\varphi \vee \psi)$ is read 'φ or ψ'.

The idea is to identify atomic propositions with propositional variables. As we have done in the example, we can identify (2.1) above with variable p_0 and (2.2) with p_1. Then, proposition (2.3) is written in CPL as $p_0 \rightarrow p_1$.

2.1.2 Semantics

The meaning of CPL formulas is given by the semantics of the logic, which uses Boolean valuations.

Definition 2.2 (Boolean Valuation)**.** A *Boolean valuation* is a function $V : \mathbb{P} \rightarrow \{0, 1\}$, which assigns a truth value to each propositional variable in $\mathbb{P}$.

The truth values 0 and 1 mean, respectively, 'false' and 'true'. Let $p \in \mathbb{P}$, $V(p) = 0$ means that the proposition represented by p is false, and it is true when $V(p) = 1$. The valuation function can be used to give the truth values to all formulas in $\mathcal{L}_{\text{CPL}}$. There are several different ways to do that. Here, we find it more convenient to use a *satisfaction relation*.

Definition 2.3 (Satisfaction Relation)**.** The *satisfaction relation* $\vDash$ between Boolean

valuations and formulas in $\mathcal{L}_{\text{CPL}}$ is recursively defined as follows:

$$
\begin{array}{lll}
V \not\models \bot & & \\
V \models p & \text{iff} & V(p) = 1 \\
V \models (\varphi \to \psi) & \text{iff} & \text{if } V \models \varphi \text{ then } V \models \psi \\
V \models \varphi \wedge \psi & \text{iff} & V \models \varphi \text{ and } V \models \psi \\
V \models \varphi \vee \psi & \text{iff} & V \models \varphi \text{ or } V \models \psi
\end{array}
$$

Example 2.1. Let the formula $p_0 \to p_1$ be given, and let V be a Boolean valuation where $V(p_0) = 1$ and $V(p_1) = 0$. We have:

$$
\begin{aligned}
V \models p_0 \to \neg p_1 \text{ iff } & \text{if } V \models p_0 \text{ then } V \models p_1 \\
& \text{iff if } V(p_0) = 1 \text{ then } V(p_1) = 1
\end{aligned}
$$

Thus, the formula $p_0 \to p_1$ is true under this Boolean valuation. ∎

When $V \models \varphi$, we say that the Boolean valuation V *satisfies* the formula φ. Boolean valuations are also called *CPL models*. When a model satisfies a formula φ, it is also called a *model of* φ. This leads to the next definition.

Definition 2.4 (Satisfiability, Validity)**.** A CPL formula φ is *satisfiable* if and only if there is a Boolean valuation V that satisfies φ. A CPL formula φ is *valid* if and only if all Boolean valuations V satisfy φ.

We use $\models \varphi$ to denote that φ is valid.

A valid CPL formula is also called a *tautology*. This notion can be generalised, as follows.

Definition 2.5 (Semantic Consequence)**.** Let Γ be a set of CPL formulas. A CPL formula φ is a *semantic consequence* of Γ if and only if every Boolean valuation that satisfies every member of Γ also satisfies φ.

We use $\Gamma \models \varphi$ to denote that φ is a semantic consequence of Γ.

Note that, if a formula $\neg\varphi$ is not valid, then there is a valuation V such that $V \not\models \neg\varphi$. This means that $V \models \varphi$. Therefore, φ is satisfiable if and only if $\not\models \neg\varphi$. Also note that $\models \varphi$ if and only if $\emptyset \models \varphi$.

For instance, the formula $p_0 \to p_1$ in Example 2.1 is satisfiable, but not valid. To see it, assume another Boolean valuation V' where $V'(p_0) = 1$ and $V'(p_1) = 0$. The reader may verify that $V' \not\models p_0 \to p_1$. As an example of valid formula, one can verify that $p_0 \to p_0$ is satisfied by all Boolean valuations.

Semantic consequence also permits us to define *equivalence* and *equisatisfiability* between formulas.

Definition 2.6 (Equivalence)**.** Two CPL formulas φ and ψ are *equivalent*, noted $\varphi \equiv \psi$, if and only if, $\varphi \models \psi$ and $\psi \models \varphi$.

Definition 2.7 (Equisatisfiability)**.** Two CPL formulas φ and ψ are *equisatisfiable*, noted $\varphi \overset{\text{sat}}{\equiv} \psi$, if and only if there is a function f between Boolean valuations such that, for all V we have:

$$V \vDash \varphi \text{ iff } f(V) \vDash \psi$$

Corollary 2.1. *$\varphi \equiv \psi$ implies $\varphi \overset{\text{sat}}{\equiv} \psi$.*

It is easy to see that the converse of Corollary 2.1 is not true. For example, $p_0 \overset{\text{sat}}{\equiv} p_1$, whereas $p_0 \not\equiv p_1$. Equivalence means that the very same models that satisfy one formula also satisfy the other, whereas equisatisfiability means that, whenever we can find a model for one formula, we can also find a model for the other one.

2.1.3 An Axiom System for CPL

In this thesis, we are often interested in deciding whether a formula is valid in a logic. Depending on the logic, there may be several automatic methods to perform this task. One of the most traditional is the one known as Hilbert-style system. Most of the time, this method is not very efficient, but it is relevant for this work, specially on chapters 4, 5 and 6. Therefore, we give it some attention here. Efficient methods for deciding whether CPL formulas are valid are discussed on Section 2.1.4. Efficient methods for deciding validity of modal logic formulas are proposed on Chapter 7.

Hilbert-style systems are also called *axiom systems*. Such a system consists of a set of *axiom schemas* $\mathcal{A}$ and a set of *inference rules* $\mathcal{R}$. An axiom schema is a "schematic" formula, in the sense that it stipulates that any formula of a certain format is an axiom. For example, assume that the formula below is an axiom schema:

$$\varphi \to (\psi \to \varphi)$$

We have that $p_0 \to (p_1 \to p_0)$ is an axiom, as well as $\neg p_0 \to ((p_1 \to p_0) \to \neg p_0)$. The former is so by setting $\varphi = p_0$ and $\psi = p_1$, whereas for the latter, we set $\varphi = \neg p_0$ and $\psi = p_1 \to p_0$. The axioms used in a derivation (see Definition 2.8) are also called *instances* of the axiom schema considered.

Inference rules have two components: a set of formulas that *triggers* the rule and a set of formulas that are *produced* by the rule, and these sets of formulas are also schemas. For example, assume the inference rule below:

$$\text{From } \varphi \text{ and } \varphi \to \psi \text{ infer } \psi$$

We have that, if the formulas p_0 and $p_0 \to p_1$ are already present in a derivation, then we can add the formula p_1 to the derivation.

A formula is considered valid in the given axiom system if one can find a *proof* for it using the axioms and inference rules available. So first, let us make precise what *proof* means.

Definition 2.8 (Derivation, Proof)**.** Let an axiom system be formed by a set of axiom schemas $\mathcal{A}$ and a set of inference rules $\mathcal{R}$. Let H be a (possibly infinite) set of formulas, called *hypotheses*. A *derivation from H* is a finite sequence of formulas $\varphi_1, \ldots, \varphi_n$ such that each formula in the sequence is either:

- a member of H,
- an axiom from $\mathcal{A}$, or
- obtained by the application of an inference rule from $\mathcal{R}$ to formulas that appear previously in the sequence.

The last formula of the sequence, φ_n, is the *derived formula*.

If the set H is empty, the derivation is called *proof*, and the last formula in the sequence is the formula that has been proved.

We use $H \vdash \varphi$ to denote that φ is derived from H and we simply use $\vdash \varphi$ to denote that φ is derived from $\emptyset$.

Example 2.2. Let us see how a derivation looks like. Recall that the formula $p_0 \to p_0$ is valid. Below, we see how it can be proved in the axiom system of Table 2.1.

1. $p_0 \to ((p_1 \to p_0) \to p_0)$	instance of $(\to 1)$
2. $(p_0 \to ((p_1 \to p_0) \to p_0)) \to$ $((p_0 \to (p_1 \to p_0)) \to (p_0 \to p_0))$	instance of $(\to 2)$
3. $(p_0 \to (p_1 \to p_0)) \to (p_0 \to p_0)$	from 1 and 2 with (RMP)
4. $p_0 \to (p_1 \to p_0)$	instance of $(\to 1)$
5. $p_0 \to p_0$	from 3 and 4 with (RMP)

Line 1 is an instance of axiom schema $(\to 1)$, where $\varphi = p_0$ and $\psi = p_1 \to p_0$. Line 2 is an instance of axiom schema $(\to 2)$, where, $\varphi = p_0$ and $\psi = p_1 \to p_0$ and $\chi = p_0$. Line 3 is obtained from the application of inference rule (RMP) to lines 1 and 2. Line 4 is another instance of axiom schema $(\to 1)$, where $\varphi = p_0$ and $\psi = p_1$. Finally, line 5 is obtained from lines 3 and 4 by the application of inference rule (RMP). ∎

Derivations are also fundamental to understand the concept of *deductive closure*, which is important, for instance, in belief revision theory, that we study in Chapter 6.

Definition 2.9 (Deductive Closure)**.** Let Γ be a set of formulas. The *deductive closure* of Γ is the set $\mathrm{Cn}(\Gamma) = \{\varphi \mid \Gamma \vdash \varphi\}$, i.e., the set of all formulas that can be derived when taking all formulas in Γ as hypotheses.

Some interesting properties follow immediately from the definitions above. First, it is easy to see that, whenever a formula can be derived from a set Γ, it can also be derived from a set Γ', where $\Gamma \subseteq \Gamma'$. We thus have the following property.

Lemma 2.2 (Monotonicity)**.** *If* $\Gamma \vdash \varphi$ *and* $\Gamma \subseteq \Gamma'$ *then* $\Gamma' \vdash \varphi$.

For the next property, assume an infinite set Γ such that $\Gamma \vdash \varphi$. Since derivations must be finite, it follows that there is only a finite set of formulas from Γ that are used on the proof of φ.

Lemma 2.3 (Compactness)**.** *If* $\Gamma \vdash \varphi$ *then there is a finite set* $\Gamma' \subseteq \Gamma$ *such that* $\Gamma' \vdash \varphi$.

$(\to 1)$	$\varphi \to (\psi \to \varphi)$	(implication 1)
$(\to 2)$	$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$	(implication 2)
$(\bot)$	$\bot \to \varphi$	(falsehood)
$(\wedge 1)$	$(\varphi \wedge \psi) \to \varphi$	(conjunction 1)
$(\wedge 2)$	$(\varphi \wedge \psi) \to \psi$	(conjunction 2)
$(\wedge 3)$	$\varphi \to (\psi \to (\varphi \wedge \psi))$	(conjunction 3)
$(\vee 1)$	$\varphi \to (\varphi \vee \psi)$	(disjunction 1)
$(\vee 2)$	$\psi \to (\varphi \vee \psi)$	(disjunction 2)
$(\vee 3)$	$(\varphi \to \bot) \to ((\psi \to \bot) \to ((\varphi \vee \psi) \to \bot))$	(disjunction 3)
$(\neg\neg)$	$((\varphi \to \bot) \to \bot) \to \varphi$	(double negation)
(RMP)	From φ and $\varphi \to \psi$ infer ψ	(modus ponens)

Table 2.1 – Axiom system of classical propositional logic

The completeness proofs in this section, as well as in Section 3 and chapters 4 and 5 use the concepts of consistent and maximal consistent sets. We therefore pay some attention to it here.

Definition 2.10 (Consistency)**.** A set of formulas Γ is *inconsistent* if and only if $\Gamma \vdash \bot$, and it is *consistent* if and only if it is not inconsistent, i.e., $\Gamma \nvdash \bot$.

Definition 2.11 (Maximal Consistent Set)**.** A set of formulas Γ is *maximal consistent* if and only if (1) it is consistent and (2) if $\Gamma \subseteq \Gamma'$ then $\Gamma = \Gamma'$.

Now, assume that a maximal consistent set Γ such that $\Gamma \vdash \varphi$. It is interesting to see that φ must be in Γ. For, if $\varphi \notin \Gamma$ were the case, then, because Γ is maximal consistent, we would have that $\Gamma \cup \{\varphi\} \vdash \bot$. But, since $\Gamma \vdash \varphi$, it is not difficult to see that one can produce a finite proof for $\bot$ using the elements of Γ. However, the latter contradicts the fact that Γ is consistent. Therefore, we have the following property.

Lemma 2.4. *Let Γ be a maximal consistent set. If $\Gamma \vdash \varphi$ then $\varphi \in \Gamma$.*

Now, the proof of the famous theorem below uses lemmas 2.2 and 2.3 above, but we spare the reader from the details.

Theorem 2.5 (Lindenbaum's Theorem)**.** *Let Γ be any consistent set. We have that $\Gamma \subseteq \Gamma^*$, for some maximal consistent set Γ^*.*

It can be shown that the axiom system in Table 2.1 is sound and complete for CPL.

Theorem 2.6 (Soundness and Completeness)**.** *Let the axiom system on Table 2.1 be given. We have that $\vDash \varphi$ if and only if $\vdash \varphi$.*

Soundness, i.e., the implication from the left to the right, is very simple. In fact, it is enough to show that every principle in Table 2.1 is valid. For the axioms, it amounts to show that they are all tautologies. For the inference rule (RMP), it amounts to show that, whenever φ and $\varphi \to \psi$ are tautologies, so is ψ. As the reader may easily verify, this is indeed the case. All this means that every formula in a proof is valid. Therefore, every proved formula is valid.

Proof of completeness, i.e., the implication from the right to the left, is much more involving. First, we show the folowing theorem.

Theorem 2.7 (Deduction Theorem)**.** *Let* $\Gamma \subseteq \mathcal{L}_{\mathrm{CPL}}$ *and* $\varphi, \psi \in \mathcal{L}_{\mathrm{CPL}}$*. If* $\Gamma \cup \{\varphi\} \vdash \psi$ *then* $\Gamma \vdash \varphi \to \psi$*. (In particular, if* $\varphi \vdash \psi$ *then* $\vdash \varphi \to \psi$*.)*

Second, we show the following lemma.

Lemma 2.8. Γ *is a maximal consistent set of formulas if and only if, for all* $\varphi \in \mathcal{L}_{\mathrm{CPL}}$*, we have:*

1. $\varphi \in \Gamma$ *iff* $(\varphi \to \bot) \notin \Gamma$.
2. $(\varphi \to \psi) \in \Gamma$ *iff, if* $\varphi \in \Gamma$ *then* $\psi \in \Gamma$.
3. $(\varphi \wedge \psi) \in \Gamma$ *iff* $\varphi \in \Gamma$ *and* $\psi \in \Gamma$.
4. $(\varphi \vee \psi) \in \Gamma$ *iff* $\varphi \in \Gamma$ *or* $\psi \in \Gamma$.

The proof of Lemma 2.8 uses Lemma 2.4 and theorems 2.5 and 2.7. As for an easy example, Axioms ($\wedge$1) and ($\wedge$2) guarantee that if $\varphi \wedge \psi \in \Gamma$, we have both $\varphi \in \Gamma$ and $\psi \in \Gamma$.

After all that, we can finally show completeness by showing the countrapositive of the claim, i.e., if a formula φ does not have a proof then it is not valid. In fact, it is possible to prove an even stronger result, called *strong completeness*, which means the following.

Theorem 2.9. $\varphi \vDash \psi$ *if and only if* $\varphi \vdash \psi$.

We finish this section with two remarks. First, as a side effect of the deduction theorem above, we have the following property.

Corollary 2.10. $\varphi \equiv \psi$ *if and only if* $\vDash \varphi \leftrightarrow \psi$.

Second, the choice of primitive connectives made here is guided by the technique used to prove completeness of the axiom system. In any case, whenever the set of primitive connectives is *adequate*, i.e., it permits to define all truth-functional operations, a sound and complete axiom system can be found.

For example, we could have chose to define CPL with negation ($\neg$) and material implication ($\to$) as primitive connectives and use abbreviations for $\bot$, $\top$, $\wedge$, $\vee$ and $\leftrightarrow$. In this case, it is shown in, e.g., (Mendelson 2015) that, the axiom system formed by ($\to$ 1), ($\to$ 2), (RMP) and the axiom schema:

($\neg$2) $(\neg\varphi \to \neg\psi) \to ((\neg\varphi \to \psi) \to \varphi)$ (negation)

is sound and complete for CPL. We choose the technique above, which uses maximal consistent sets and the Lindenbaum's Theorem, because this is the technique commonly used to show completeness of axiom systems in modal logics. We will therefore build axiom systems for modal logics from that.

Deductive closures and maximal consistent sets are also usefull in some other contexts. For example, knowledge bases inconsistency mesures (De Bona et al. 2018; Hunter and Konieczny 2010; Mu et al. 2011) are havily based on this notion.

2.1.4 Automated Reasoning in CPL

Let a formula $\varphi \in \mathcal{L}_{\text{CPL}}$ be given, the problem of deciding whether φ is satisfiable in a logic is called *satisfiability checking problem*. In the case of CPL, Cook (1971) proved that any recognition problem that can be solved in polynomial time by a non-deterministic Turing machine can be reduced to the CPL satisfiability checking problem. In other words, CPL satisfiability checking is a NP-Hard problem. Since there are several algorithms that decide satisfiability of CPL formulas in polynomial time in a non-deterministic Turing machine, we have that CPL satisfiability checking problem is in NP. Altogether, this means that CPL satisfiability checking is a NP-Complete problem.

Note that, an algorithm deciding satisfiability of CPL formulas can easily be turned into an algorithm deciding validity of CPL formulas and vice-versa. Indeed, because φ is satisfiable if and only if $\neg\varphi$ is not valid, we immediately have that *CPL validity checking* is a co-NP-complete problem.

As said before, axiom systems are not efficient automatic methods for reasoning in CPL. This is why, in this subsection, we show two methods for CPL satisfiability checking that are more efficient in practice. The tableau method and the CDCL algorithm.

The Tableau Method

One of the simplest algorithms deciding satisfiability (thus, also validity) of CPL formulas is the *tableau method*. (For a thorough exposition of the subject the reader may consult (D'Agostino et al. 1999).) The idea of this algorithm is to decompose the formula given as input in sub-formulas until it reaches its atoms (constants or variables) and then build-up a Boolean valuation for them.

But, before presenting the method, we need to deal with a tiny language problem. In fact, it turns out that it is easier to design a tableau method for CPL when negation ($\neg$), conjunction ($\wedge$) and disjunction ($\vee$) are taken as the primitive connectives of the language, and the other ones are defined as abbreviations. Note that this can be done without affecting the expressiveness of the logic. In the following, we assume that $\mathcal{L}_{\text{CPL}}$ is defined in this way.

The tableau method explores the sub-formulas of the formula φ that is given it as input. So, before presenting the method, we need to make precise what sub-formula means.

Definition 2.12 (Set of Sub-Formulas)**.** Let $\varphi \in \mathcal{L}_{\mathrm{CPL}}$, $\mathrm{sub}(\varphi)$ denotes the *set of sub-formulas* of φ, which is defined recursively, as follows:

$$\begin{aligned}
&\mathrm{sub}(\bot) = \{\bot\} \\
&\mathrm{sub}(p) = \{p\} \\
&\mathrm{sub}(\neg\varphi) = \{\neg\varphi\} \cup \mathrm{sub}(\varphi) \\
&\mathrm{sub}(\varphi \circ \psi) = \{\varphi \circ \psi\} \cup \mathrm{sub}(\varphi) \cup \mathrm{sub}(\psi) \qquad \text{where } \circ \in \{\to, \wedge, \vee\}
\end{aligned}$$

It is important for the complexity proof of the method to see that the number of elements of the set $\mathrm{sub}(\varphi)$ is proportional to the *length* of the formula φ. The latter is defined as follows.

Definition 2.13 (Length)**.** The *length* of a formula $\varphi \in \mathcal{L}_{\mathrm{CPL}}$ is the natural number $\mathrm{len}(\varphi)$ defined recursively, as follows:

$$\begin{aligned}
&\mathrm{len}(\bot) = 1 \\
&\mathrm{len}(p) = 1 \\
&\mathrm{len}(\neg\varphi) = 1 + \mathrm{len}(\varphi) \\
&\mathrm{len}(\varphi \circ \psi) = 1 + \mathrm{len}(\varphi) + \mathrm{len}(\psi) \qquad \text{where } \circ \in \{\to, \wedge, \vee\}
\end{aligned}$$

It is easy to see that the number of elements of $\mathrm{sub}(\varphi)$ equals $\mathrm{len}(\varphi)$.
We can finally present the definition of a tableau for a CPL formula.

Definition 2.14 (Tableau)**.** Let φ be a CPL formula. A *tableau for* φ is a non-empty set $T \subseteq \mathrm{sub}(\varphi)$ such that $\varphi \in T$ and T satisfies the following conditions:

1. $\bot \notin T$.
2. $\varphi \in T$ if and only if $\neg\varphi \notin T$.
3. if $\neg\neg\psi \in T$ then $\psi \in T$.
4. if $\psi_1 \wedge \psi_2 \in T$ then $\psi_1 \in T$ and $\psi_2 \in T$.
5. if $\psi_1 \vee \psi_2 \in T$ then $\psi_1 \in T$ or $\psi_2 \in T$.

Lemma 2.11. *Let $\varphi \in \mathcal{L}_{\mathrm{CPL}}$. There is a tableau for φ if and only if φ is satisfiable.*

To prove the implication from the right to the left of Lemma 2.11, we assume a satisfiable formula φ and show that there is an algorithm (Algorithm 2.1) that constructs a tableau T for it. To prove the implication from the left to the right, we assume a tableau T for φ. From T, we construct a Boolean valuation and, using an induction on the structure of φ, we conclude that this Boolean valuation satisfies φ.

The procedure described in Algorithm 2.1 tries to construct a tableau for the input formula φ. It starts with the singleton $T = \{\varphi\}$ and decomposes φ by adding its sub-formulas to T, aiming at satisfying all conditions in Definition 2.14. For example, if $(\psi_1 \wedge \psi_2) \in T$, then it adds both ψ_1 and ψ_2 to T. (The implemented procedure also removes $(\psi_1 \wedge \psi_2)$ from T to avoid performing this step again.) When the procedure

input: The set $T = \{\varphi\}$
output: **true** if φ is satisfiable, **false** otherwise

```
1  function tableau(T)
2      if ⊥ ∈ T or {ψ, ¬ψ} ⊆ T then
3          return false
4      if ¬¬ψ ∈ T then
5          return tableau(T \ {¬¬ψ} ∪ {ψ})
6      if (ψ1 ∧ ψ2) ∈ T then
7          return tableau(T \ {ψ1 ∧ ψ2} ∪ {ψ1, ψ2})
8      if (ψ1 ∨ ψ2) ∈ T then
9          return tableau(T \ {ψ1 ∨ ψ2} ∪ {ψ1}) or tableau(T \ {ψ1 ∨ ψ2} ∪ {ψ2})
10     return true
```

Algorithm 2.1: Tableau method for classical propositional logic

faces a choice, it branches. For example, if $(\psi_1 \vee \psi_2) \in T$, then it creates two different tableaux T' and T''. It adds all elements of T plus ψ_1 to the first tableau T', and adds all elements of T plus ψ_2 to the second tableau T''. (Again, it removes $(\psi_1 \vee \psi_2)$ from T to avoid repeating this step.) Eventually, the procedure reaches a state where either it violates one of the conditions in Definition 2.14, or the input formula φ has been completely decomposed. In the first case, it returns **false**, meaning that no Boolean valuation satisfying the formulas in the corresponding tableau T can be found. In the second case, it returns **true**, meaning that a Boolean valuation satisfying all formulas in T is possible.

For the branches where the procedure returns **true**, one can construct a Boolean valuation that satisfies φ, as follows. For all variables p that occur in φ, $V(p) = 1$ if p is in the branch, otherwise $V(p) = 0$. By using Lemma 2.11, one can show that such a valuation satisfies the input formula φ.

An example of the execution of Algorithm 2.1 is depicted in Figure 2.1. The recursive calls of function tableau() form the tree displayed in the figure. In the root of the tree, the tableau T only contains the input formula. In its child, it contains both conjuncts. In the subsequent step, the procedure creates two tableaux T' and T'', each one containing one of the disjuncts of the formula $\neg p_0 \vee p_1$. Tableau T' contains two contradictory formulas, this is why this branch is stopped and *closed*, meaning that this branch can be discarded. The branch with tableau T'' also stops, but it does not contain contradictions. Therefore, one can create a Boolean valuation V that satisfies the input formula φ, as described before. Since both p_0 and p_1 are in T'', we have $V(p_0) = 1$ and $V(p_1) = 1$.

Termination of the procedure is guaranteed by the finiteness of $\mathrm{sub}(\varphi)$. It is also not hard to show that the depth of the tree is bound by the size of $\mathrm{sub}(\varphi)$. Altogether, this means that we can show that the maximal number of steps executed by the procedure is bounded by a polynomial function on $\mathrm{len}(\varphi)$, when executed in a non-deterministic Turing machine. Therefore, the tableau method is optimal, with respect to complexity

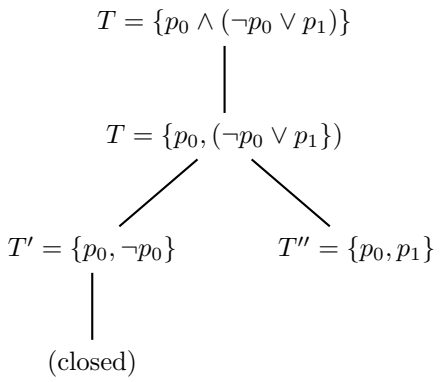


Figure 2.1 – Execution of the tableau method for $p_0 \wedge (\neg p_0 \vee p_1)$

class of the problem.

The CDCL Algorithm

The tableau method presented on the previous section is optimal, but it is not the most performant method. The most efficient software deciding satisfiability of CPL formulas available today are based on an algorithm called *conflict driven clause learning (CDCL)* (Bayardo and Schrang 1997; Marques-Silva and Sakallah 1996, 1999). This algorithm uses several different techniques to speed up the process of finding a model for a formula. It is an evolution of the DPLL algorithm (Davis et al. 1962; Davis and Putnam 1960), where the main differences from the latter are (1) its conflict analysis, which prevents the algorithm from taking a decision that has shown to be a bad one on previous iterations; and (2) its capacity to perform non-chronological backtracks.

CDCL takes as input CPL formulas in a special format called *conjunctive normal form*. Thus, we need to define it first.

Definition 2.15 (Literals, Clauses, Negation and Conjunctive Normal Forms)**.** A *literal* is a propositional variable or a negation of a propositional variable.

A *clause* is a disjunction of literals.

A CPL formula φ is in *negation normal form (NNF)* if and only if the negations in φ appear only immediately in front of propositional variables.

A CPL formula is in *conjunctive normal form (CNF)* if and only if it is a conjunction of clauses.

Any formula $\varphi \in \mathcal{L}_{\text{CPL}}$ can be translated into an equivalent formula in CNF. For example, the following formulas are not in CNF:

$$\neg(p_0 \wedge p_1) \qquad \neg(p_0 \vee p_1) \qquad \neg\neg p_0 \qquad \neg(p_0 \wedge (\neg p_1 \vee p_2))$$

whereas the following ones are their equivalent in CNF:

$$\neg p_0 \vee \neg p_1 \qquad \neg p_0 \wedge \neg p_1 \qquad p_0 \qquad (\neg p_0 \vee p_1) \wedge (\neg p_0 \vee \neg p_2)$$

Any formula $\varphi \in \mathcal{L}_{\text{CPL}}$ can be translated into an equivalent formula in NNF in linear time and space (Baaz et al. 2001). However, in general, the translation of a CPL formula into an equivalent one in CNF generates an exponentially larger formula. However, if we allow the introduction of new propositional variables, it is possible to generate an equisatisfiable formula in CNF such that its length is bound by a polynomial function on the length of the original formula. Moreover, the algorithm producing such formula runs in polynomial time (Tseitin 1983). Therefore, it is safe to assume an algorithm that take formulas in CNF as input.

CDCL uses *unit propagation*. This is a technique that simplifies CNF formulas by using the following two properties of CPL.

Theorem 2.12 (Resolution (Robinson 1965))**.** *Let* $\varphi, \psi, \chi \in \mathcal{L}_{\text{CPL}}$. *We have* $\{(\varphi \vee \psi), (\neg\varphi \vee \chi)\} \vDash (\psi \vee \chi)$

Theorem 2.13. *Let* $\varphi, \psi \in \mathcal{L}_{\text{CPL}}$. $\{\varphi, (\varphi \vee \psi)\} \vDash \varphi$.

Theorem 2.12 means that, if $(\psi \vee \chi)$ is not satisfiable then $(\varphi \vee \psi) \wedge (\neg\varphi \vee \chi)$ is also not satisfiable. Therefore, in order to decide satisfiability for the latter, it is enough to decide satisfiability for the former. Note that it is better to use the former formula, because it is smaller. Theorem 2.13 is analogous.

When the input formula φ has a unit clause (i.e., a clause with only one literal) this means that this literal must be assigned to true in the Boolean valuation that satisfies the entire formula. In other words, if the literal is a variable p then $V(p)$ must be 1, and if the literal is a negation of a variable p then $V(p)$ must be 0. Therefore, there is no decision to be taken with respect to this literal. Once this is done, we can propagate this information across the formula, hence the unit propagation.

Example 2.3. To see unit propagation in practice, consider the following CPL formula in CNF:

$$(p_0 \vee p_1) \wedge (\neg p_0 \vee p_2) \wedge (\neg p_2 \vee p_3) \wedge p_0$$

We will propagate the unit clause p_0. We have:

1. $(p_0 \vee p_1) \wedge (\neg p_0 \vee p_2) \wedge (\neg p_2 \vee p_3) \wedge p_0$
2. $(\neg p_0 \vee p_2) \wedge (\neg p_2 \vee p_3) \wedge p_0$
3. $p_2 \wedge (\neg p_2 \vee p_3) \wedge p_0$

Line 2 is obtained from Line 1 by removing the first conjunct (Theorem 2.13). Line 3 is obtained from Line 2 by removing the first conjunct again (Theorem 2.12). Note that we now have a new unit clause p_2, which can be propagated. Thus, unit propagation continues. We have:

4. $p_2 \wedge p_3 \wedge p_0$

which is obtained from Line 3 by removing the second conjunct (Theorem 2.12). Now, all clauses are unit, but there is no non-unit clauses to propagate. Unit propagation stops. Note that it is much easer to find a model for the formula on Line 4, than for the one on Line 1. ∎

The second important technique used by CDCL is the *conflict analysis*. Once unit propagation stops, if there are still non-unit clauses, it is time to take decisions, i.e., we must arbitrarily assign a truth value to some variable and then check whether the Boolean valuation we are building still satisfies all the clauses. Sometimes, we take a decision and we find out that the Boolean valuation does not satisfy some clause. This is called *conflict*. Once a conflict is detected, CDCL does two things. First, it adds a new clause to the formula, which will prevent this conflict from happen again, this is called *clause learning*. Second, it tries to find the decision that most likely lead to the conflict and restart from there, i.e., it *backtracks*. The explanation of clause learning and backtrack techniques is involving. We do not explain them in detail here. For a through exposition, the reader can consult (Marques-Silva et al. 2009). In the sequel we only explain roughly how clause learning works.

During its execution, CDCL builds a graph that contains the decisions taken and their consequences. When a conflict is detected, one can build, from this graph, a clause that contains information to prevent the same chain of decisions to be taken again. For example, consider the formula:

$$(\neg p_4 \vee \neg p_5) \wedge (p_{21} \vee \neg p_4 \vee \neg p_6) \wedge (p_5 \vee p_6)$$

There are no unit clauses to propagate. So, CDCL must take some decisions. Suppose it makes the assignments $V(p_{21}) = 0$ and $V(p_4) = 1$. Once it is done, the formula can be simplified. We now have:

$$\neg p_5 \wedge \neg p_6 \wedge (p_5 \vee p_6)$$

We now can apply unit propagation to $\neg p_5$. We have:

$$p_6 \wedge \neg p_6$$

which is a conflict. This means that the same decisions should not be taken again. In this example, CDCL will learn the clause $p_{21} \vee \neg p_4$, which is somewhat the contrary of what we have done in the beginning. This is added to the input formula and the algorithm backtracks. A different decision will be taken and maybe this time a model will be found.

We can now present the outline of CDCL in Algorithm 2.2. It starts by performing unit propagation. After that, CDCL chooses a variable, assigns a truth value to it and performs unit propagation again. If a conflict is detected, it is analysed, then clauses are learned, and then CDCL decides to which level it will perform the backtrack. The process repeats until a Boolean valuation satisfying the formula is found, in which case it returns 'true'. Otherwise, it returns 'false'.

We note that current implementations of CDCL bring together a number of additional improvements. Among them, we can mention:

- use of lazy data structures for the representation of formulas;
- periodically restarting backtrack search;
- deletion policies for learnt clauses (to avoid overflow the memory);
- etc.

input: A CPL formula φ in CNNF and a Boolean valuation V
output: **true** if φ is satisfiable, **false** otherwise

```
1  function cdcl(φ, V)
2      if unit_propagation(φ, V) = conflict then
3          return false
4      dl := 0
5      while not all_vars_assigned(φ, V) do
6          (p, v) := pick_branching_variable(φ, V)
7          dl := dl + 1
8          V := V ∪ {(p, v)}
9          if unit_propagation(φ, V) = conflict then
10             β := conflict_analysis(φ, V)
11             if β < 0 then
12                 return false
13             else
14                 backtrack(φ, V, β)
15                 dl := β
16     return true
```

Algorithm 2.2: The CDCL algorithm

2.2 Belief Revision Theory

In Chapter 6, we propose a framework for modeling belief change in multi-agent scenarios. Therefore, we make a very brief introduction to belief revision here. For more information, the reader my consult, e.g., (Gärdenfors 2008).

A simple way to represent agents beliefs is to use a set containing the sentences believed by the agent. This set is commonly called a *belief set.* The sentences in the belief set must be written in some language. We normally chose a formal language, because the meanings of such sentences as well as the relations among them can be defined more precisely. If we chose $\mathcal{L}_{\text{CPL}}$, a belief set is defined as follows.

Definition 2.16 (Belief Set)**.** A belief set is a deductively closed set of formulas in CPL.

Let K be a consistent belief set. For any sentence $\varphi \in \mathcal{L}_{\text{CPL}}$, three different epistemic attitudes concerning φ can be expressed:

Accepted: $\varphi \in K$.

Rejected: $\neg\varphi \in K$.

Indetermined: $\varphi \notin K$ and $\neg\varphi \notin K$.

When an agent changes her epistemic attitude concerning φ, this means that the agent goes from one of these attitudes to another. Therefore, there are in total six

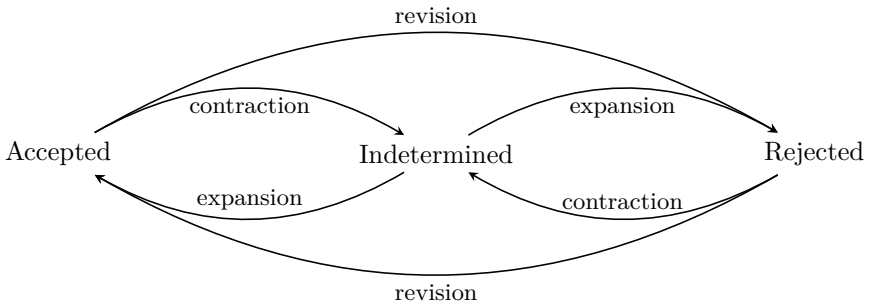


Figure 2.2 – Types of belief change

different ways of changing epistemic attitudes. These changes are grouped into three different types (see also Figure 2.2):

Expansion: φ goes from indetermined to either accepted or rejected.

Contraction: φ goes from accepted or rejected to indetermined.

Revision: φ goes either from accepted to rejected or it goes from rejected to accepted.

2.2.1 Expansion

The first type of attitude change is expansion. This can be seen as the event of the agent "learning" a new information that she was not aware of. For instance, suppose that φ is indetermined for the agent. An expansion by φ makes the agent accept that φ (and thus reject $\neg\varphi$). An expansion by $\neg\varphi$ makes the agent reject φ (and thus accept $\neg\varphi$).

Note that the expansion of K by φ is not as simple as $K \cup \{\varphi\}$. This is because the latter must also be a belief set, which means that it must be deductively closed. In other words, once φ is added to the belief set, all formulas that can be derived with this new addition must also be added to the resulting belief set. In the particular case where φ contradicts some formula already present in K, the contradiction $\bot$ is added to the resulting belief set. In this case, the resulting belief set should be trivial, i.e., $(K + \varphi) = \mathcal{L}_{\text{CPL}}$.

For example, suppose that $(p_0 \to p_1) \in K$. If we expand K by p_0, then p_0 is added to K as well as p_1. If it were the case that $\neg p_1 \in K$, then we have, as a result, $K = \mathcal{L}_{\text{CPL}}$.

Alchourrón, Gärdenfors and Makinson (Alchourrón et al. 1985; Gärdenfors 2008) proposed postulates for the expansion, contraction and revision of belief sets. These postulates logically encode the constraints expected on the behaviour of such operators.

The postulates for expansion are the following. Let K be a belief set and $\varphi \in \mathcal{L}_{\text{CPL}}$.

(K + 1) $K + \varphi$ is a belief set	(closure)
(K + 2) $\varphi \in K + \varphi$	(success)
(K + 3) $K \subseteq K + \varphi$	(inclusion)
(K + 4) If $\varphi \in K$ then $K + \varphi = K$	(vacuity)
(K + 5) If $K \subseteq K'$ then $K + \varphi \subseteq K' + \varphi$	(monotony)
(K + 6) $K + \varphi$ is the smallest belief set satisfying (K + 1)–(K + 5)	(minimality)

Postulate (K + 1) stipulates that the resulting of expanding a belief set is also a belief set. Postulate (K + 2) stipulates that φ will be accepted after expanding the belief set by φ. Postulate (K + 3) stipulates that nothing is lost with the addition of φ to the belief set. Postulate (K + 4) stipulates that K does not change if φ is already in K. For postulate (K + 5), assume that K' contains more beliefs than K. In this case, there is no reason for the expansion of K by φ to contain more beliefs than the expansion of K' by the same formula φ. Postulate (K + 6) stipulates that, as its name suggests, the change caused in K by the expansion of φ is minimal.

It turns out that there is only one expansion operator + satisfying these postulates. As stated by the result below.

Theorem 2.14 (Representation Theorem)**.** *The expansion function* + *satisfies* (K + 1)–(K + 6) *if and only if* $K + \varphi = \text{Cn}(K \cup \{\varphi\})$.

Theorem 2.14 is an important result. Not only the expansion operator + exists, but it is unique. In addition, it seems to be a very intuitive operation of simply adding a new formula to the belief set and then close it deductively.

2.2.2 Contraction

The second type of change is contraction. This can be seen as the event of the agent "forgetting" a piece of information. For example, suppose that the agent accepts φ (and thus rejects $\neg\varphi$). A contraction by φ makes both φ and $\neg\varphi$ become for the agent indetermined, i.e., the agent is not sure of the truth value of φ anymore. Now, suppose the opposite, that the agent rejects φ (and thus she accepts $\neg\varphi$). In this case, a contraction by $\neg\varphi$ makes $\neg\varphi$ become indetermined for the agent.

Analogously as to expansion, the contraction of K by φ is not as simple as $K \setminus \{\varphi\}$. All formulas that "need" φ to be derived must also be removed from the resulting belief set. In addition, all formulas from which one can derive φ must also be removed. In the particular case where $\vDash \varphi$, its subtraction subtracts all formulas from the resulting belief set. In this case, the resulting belief set is $(K - \varphi) = \emptyset$.

For example, suppose that $\{p_0, p_1, (p_0 \to p_1)\} \subseteq K$. If we contract K by p_1, then p_1 is removed from K which means that either p_0 or $p_0 \to p_1$ must also be removed. Note that, as usual in contraction, in this case we have a choice.

The postulates for contraction are the following.

$(\mathrm{K}-1)$ $K-\varphi$ is a belief set	(closure)
$(\mathrm{K}-2)$ $K-\varphi \subseteq K$	(inclusion)
$(\mathrm{K}-3)$ If $\varphi \notin K$ then $K-\varphi = K$	(vacuity)
$(\mathrm{K}-4)$ If $\nvdash \varphi$ then $\varphi \notin K-\varphi$	(success)
$(\mathrm{K}-5)$ If $\varphi \in K$ then $K \subseteq (K-\varphi)+\varphi$	(recovery)
$(\mathrm{K}-6)$ If $\vdash \varphi \leftrightarrow \psi$ then $K-\varphi = K-\psi$	(extensionality)
$(\mathrm{K}-7)$ $K-\varphi \cap K-\psi \subseteq K-(\varphi \wedge \psi)$	(intersection)
$(\mathrm{K}-8)$ If $\varphi \notin K-(\varphi \wedge \psi)$ then $K-(\varphi \wedge \psi) \subseteq K-\varphi$	(conjunction)

As for expansion, the first postulate stipulates that the result of a contraction is a belief set. Postulate $(\mathrm{K}-2)$ stipulates that no new information should be added to K by a contraction. Postulate $(\mathrm{K}-3)$ stipulates that K should not be changed if φ does not belong to it. Postulate $(\mathrm{K}-4)$ stipulates that φ does not belong to the contraction by φ, unless it is valid. Postulate $(\mathrm{K}-5)$ stipulates that all beliefs in K are recovered if first contracting and then expanding again by the same formula. Postulate $(\mathrm{K}-6)$ stipulates that a contraction by an equivalent formula should give the same result. Postulates $(\mathrm{K}-7)$ and $(\mathrm{K}-8)$ deal with the contraction by a conjunction $\varphi \wedge \psi$. Without entering in too much detail, they stipulate that a choice between φ and ψ must be made, and what must be taken into account when making this choice.

2.2.3 Revision

The third type of change is revision. This can be seen as the event of the agent "changing her mind" about a piece of information. Suppose that the agent accepts φ and rejects $\neg\varphi$. A revision by $\neg\varphi$ makes the agent accept $\neg\varphi$ and reject φ. Analogously, if the agent accepts $\neg\varphi$ and rejects φ, a revision by φ makes the agent accept φ and reject $\neg\varphi$.

Again, if we revise a belief set K by φ, it is not as simple as just add φ to the belief set. All formulas that are implied by φ must be added and also, all formulas that are implied by $\neg\varphi$ must be removed.

The postulates for revision are the following.

$(\mathrm{K}*1)$ $K*\varphi$ is a belief set	(closure)
$(\mathrm{K}*2)$ $\varphi \in K*\varphi$	(success)
$(\mathrm{K}*3)$ $K*\varphi \subseteq K+\varphi$	(inclusion)
$(\mathrm{K}*4)$ If $\varphi \in K$ then $K+\varphi \subseteq K*\varphi$	(vacuity)
$(\mathrm{K}*5)$ $\bot \in K*\varphi$ if and only if $\vdash \neg\varphi$	(coherence)
$(\mathrm{K}*6)$ If $\vdash \varphi \leftrightarrow \psi$ then $K*\varphi = K*\psi$	(extensionality)
$(\mathrm{K}*7)$ $K*(\varphi \wedge \psi) \subseteq (K*\varphi)+\psi$	(conjunctive inclusion)
$(\mathrm{K}*8)$ If $\neg\psi \notin K*\varphi$ then $(K*\varphi)+\psi \subseteq K*(\varphi \wedge \psi)$	(conjunctive vacuity)

Postulates (K $*$ 1) and (K $*$ 2) play the same role as for expansion and contraction. Postualtes (K $*$ 3) and (K $*$ 4) together stipulate that, when $\neg\varphi \notin K$, the revision by φ is just an expansion. Postualte (K $*$ 5) stipulates that a revision by φ trivialises only when φ is an inconsistent formula. Postulate (K $*$ 6) stipulates that the revision operator should perform the same operation for equivalent formulas. Postulates (K $*$ 7) and (K $*$ 8) stipulate the behaviour of the revision operator with conjunctions.

The following definition and theorem confirm the intuition behind revision mentioned before and depicted in Figure 2.2, namely, that revision is a contraction followed by an expansion.

Definition 2.17 (Levi Identity (Levi 1977))**.** $K * \varphi = (K - \neg\varphi) + \varphi$

Theorem 2.15. *If the contraction function $-$ satisfies* (K $-$ 1)–(K $-$ 4) *and* (K $-$ 6) *and the expansion $+$ satisfy* (K $+$ 1)–(K $+$ 6)*, then the revision function $*$ obtained from Definition 2.17 satisfy* (K $*$ 1)–(K $*$ 6)*.*

Theorem 2.16. *Suppose that the assumptions of Theorem 2.15 are fulfilled. Then we have:*

1. *If* (K $-$ 7) *is satisfied then the revision function $*$ satisfies* (K $*$ 7)*.*
2. *If* (K $-$ 8) *is satisfied then the revision function $*$ satisfies* (K $*$ 8)*.*

Another important important result is given in the sequel.

Definition 2.18 (Harper Identity (Harper 1976))**.** $K - \varphi = K \cap K * \neg\varphi$

Theorem 2.17. *If the revision function $*$ satisfies* (K $*$ 1)–(K $*$ 6) *then the contraction function obtained from Definition 2.18 satisfies* (K $-$ 1)–(K $-$ 6)*.*

Theorem 2.18. *Suppose that the revision function satisfies* (K $*$ 1)–(K $*$ 6)*, then:*

1. *If* (K $*$ 7) *is satisfied then the revision function $*$ satisfies* (K $-$ 7)*.*
2. *If* (K $*$ 8) *is satisfied then the revision function $*$ satisfies* (K $-$ 8)*.*

Levi identy above shows that belief revision can be defined using contraction and expansion. This means that, when faced with a piece of information φ that contradicts her beliefs, the agent can contract by $\neg\varphi$ and then expand by φ to integrate it.

Harper identity above shows us that a contraction by φ is an intersection with the revision by $\neg\varphi$.

2.3 Conclusion

In the first part of this chapter we saw a brief introduction to classical propositional logic. We saw its syntax and semantics, but also some basic definitions such as derivations, deductive closure, maximal consistent sets and an axiom system. In the second part, we saw the very basics of belief revision theory, the classical AGM postulates, Levi and Harper identities. This material will be useful on Chapter 6, when expansion and revision are be extended to multi-agent settings.

In the next chapter, we will see a brief introduction to modal logic, which will in turn be referred to in all subsequent chapters.

Chapter 3

Modal Logic

The logics used in this thesis are all *modal logics*. This is a family of logics that use *modal operators* to express a variety of notions, such as knowledge, belief, preference, obligation, action and time. In this work, we see several modal logics, such as *epistemic logics*, *deontic logic*, *propositional dynamic logic*, *alternating-time temporal logic* as well as some new ones. All of them follow the same principle: they extend classical propositional logic by some modal operators, which are used to express those different notions.

Because of this extensive use of modal logics, this chapter is dedicated to an introduction to modal logic. This introduction is, of course, not meant to be complete. The interested reader may find more on theses subjects by consulting very good textbooks, such as (Blackburn et al. 2007, 2001; Chellas 1980; Hugues and Cresswell 1996)

3.1 introduction

Modal logic (ML) extends classical propositional logic, in the sense that it is used to represent and reason about modal propositions. For example, the sentence below is a modal proposition:

it is possible that John teaches at the university.

In modal logic notation, this is represented by $\Diamond p_0$. As before, p_0 represents the proposition 'John teaches at the university', but here, in addition, we have $\Diamond$ representing 'it is possible'. The symbol $\Diamond$ is used to qualify the truth value of the sentence. This is why it is called a *modality*, or a *modal operator*. The above modal proposition is true if and only if there is a situation (or, it is possible to imagine a situation) where p_0 is true. Note that its truth value does not depend on the actual truth value of the proposition p_0. This is why modal logic is considered to be more expressive than classical propositional logic. Indeed, the meaning of operator $\Diamond$ cannot be expressed by any composition of

CPL connectives.[1]

CPL operators are also allowed in modal logic. For example, the formula:

$$\Diamond p_0 \wedge \Diamond \neg p_1$$

is a well formed modal proposition meaning 'it is possible that John works at the university and it possible that John is an academic'. Another interesting example is the well formed modal proposition:

$$\neg \Diamond \neg p_0 \tag{3.1}$$

which means 'it is not possible that John does not teach at the university'. Following the intuitive meaning given before, this modal proposition is true if and only if there is no situation where p_0 is false. In other words, it is true if and only if p_0 is true in all (conceivable) situations. Hence, we conclude that it is equivalent to the sentence:

it is necessary that John teaches at the university.

The modality 'it is necessary' is also present in modal logic. It is expressed by the symbol $\Box$. Indeed, in modal logic, the modal proposition:

$$\Box p_0 \tag{3.2}$$

is equivalent to (3.1).

In the sequel, we present the formal definition of modal logic. We present only the group of modal logics called *normal modal logics*. These are modal logics based on the axiom system K (which is explained latter on this chapter). The other, non-normal, modal logics are also interesting, but we do not need to study them to understand the work presented in this thesis.

In order to be general, we present the definition of multi-modal logic. This is modal logic with several different modal operators. Each modal operator represents a different modality.

Maybe, the simplest way to understand this is to consider different agents (persons, robots, computers, etc.) having different "points of view". For example, for the first agent, which we call m_0, John teaches at the univerisity, whereas for the second agent, which we call m_1, John may not teach at the university. We have:

$$[m_0]p_0 \wedge \langle m_1 \rangle \neg p_0$$

Here, $[m_0]$ is a modal operator representing 'it is necessity for m_0', and $\langle m_1 \rangle$ is a modal operator representing 'it is possible for m_1'. Then, this modal proposition means 'it is necessary for m_0 (the first agent) that John teaches at the university and it is possible for m_1 (the second agent) that John teaches at the university'.

[1] In fact, we must be careful about what 'more expressive' really means. We will see in Chapter 7 that it is possible to translate formulas in ML into formulas in CPL. The translation requires the addition of new propositions in the formula. Moreover, the translated formula in CPL is, in general, exponentially larger than the original ML formula. In this case, this unavoidable "exponential blowing up" is another way to justify the claim that modal logic is "more expressive" than classical propositional logic.

3.2 Syntax

Definition 3.1 (Vocabulary)**.** A *vocabulary* is a pair $\langle \mathbb{P}, \mathbb{M} \rangle$, where $\mathbb{P}$ is a countable (possibly infinite) set of propositional variables, and $\mathbb{M}$ is a finite set of modal contexts.

Definition 3.2 (Formula)**.** Let a vocabulary $\langle \mathbb{P}, \mathbb{M} \rangle$ be given. The *language of modal logic*, also called the *set of modal logic formulas* is noted $\mathcal{L}_{\text{ML}}$ and is defined by the following grammar in Bakus-Nahur form:

$$\varphi ::= \bot \mid p \mid (\varphi \to \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid ([m]\varphi) \mid (\langle m \rangle \varphi)$$

where p is any variable from $\mathbb{P}$, and m is any modal context from $\mathbb{M}$.

We define the same abbreviations as for $\mathcal{L}_{\text{CPL}}$ (Definition 2.1). In addition, we sometimes use $[m]^n$ to abbreviate a sequence of n operators $[m]$ (i.e., $[m]^0\varphi \stackrel{\text{def}}{=} \varphi$ and $[m]^{n+1}\varphi \stackrel{\text{def}}{=} [m][m]^n\varphi$, where $n \geq 0$), and analogously for $\langle m \rangle$.

As for CPL, we sometimes omit outer parentheses for convenience.

When we use a language that has only one modal operator, we sometimes prefer to write modal formulas using $\Box$ and $\Diamond$, because they are more readable than their multi-modal counterparts $[m]$ and $\langle m \rangle$. Thus, whenever formulas of the form $\Box\varphi$ and $\Diamond\varphi$ are used, they mean, respectively, $[m_0]\varphi$ and $\langle m_0 \rangle \varphi$, and the set of modal contexts of the corresponding language is $\mathbb{M} = \{m_0\}$.

The formula $[m]\varphi$ is read 'box m φ', and the formula $\langle m \rangle \varphi$ is read 'diamond m φ'. Their interpretations vary according to the application. One of the most classical applications of modal logic is the one seen in the beginning of this chapter. It is used to model sentences of the form 'it is necessary that φ' and 'it is possible that φ'. The operator $[m]$ is then called the necessity operator and $\langle m \rangle$ is called the possibility operator.

We use the same definitions of literals, clauses and CNF for modal logic formulas (Definition 2.15). But note that, in general, formulas in $\mathcal{L}_{\text{ML}}$ containing modal operators cannot be efficiently translated to formulas in CNF.[2]

The *length* of modal formulas is defined in the same way as for CPL formulas (Definition 2.13) plus:[3]

$$\text{len}([m]\varphi) = \text{len}(\langle m \rangle \varphi) = 1 + \text{len}(m) + \text{len}(\varphi)$$

The set of sub-formulas and negations of sub-formulas in ML is defined in the same way as for CPL (Definition 2.12) plus:

$$\text{sub}(\circ\varphi) = \{\circ\varphi\} \cup \text{sub}(\varphi) \quad \text{where } \circ \in \{[m], \langle m \rangle\}$$

[2] In fact, Hugues and Cresswell (1996) define the *modal conjunctive normal form (MCNF)*. However, this normal form is not a canonical form, in the sense that every formula can be translated to it, for all modal logics. Therefore, we do not use it here.

[3] Note that $\text{len}(m)$ may be greater than 1 when m is, as in Chapter 5, something more complex than a simple modal context.

Definition 3.3 (Modal Degree)**.** Let $\varphi \in \mathcal{L}_{\mathrm{ML}}$, $\mathrm{md}(\varphi)$ denotes the *modal degree* of φ, which is defined recursively, as follows:

$$\begin{aligned}
&\mathrm{md}(\top) = \mathrm{md}(p) = 0\\
&\mathrm{md}(\neg\varphi) = \mathrm{md}(\varphi)\\
&\mathrm{md}(\varphi \wedge \psi) = \mathrm{md}(\varphi \vee \psi) = \max(\mathrm{md}(\varphi), \mathrm{md}(\psi))\\
&\mathrm{md}([m]\varphi) = \mathrm{md}(\langle m\rangle\varphi) = 1 + \mathrm{md}(\varphi)
\end{aligned}$$

3.3 Semantics

The semantics of normal modal logics is often defined using Kripke models (named after Kripke (1959), who originally proposed it). They are formally defined as follows.

Definition 3.4 (Kripke model)**.** Let a vocabulary $\langle \mathbb{P}, \mathbb{M}\rangle$ be given. A *Kripke model* is a triplet $\langle W, R, I\rangle$, where:

- W is a non-empty set of possible worlds (sometimes called states, or situations);
- $R : \mathbb{M} \to \wp(W \times W)$ associates an accessibility relation $R(m)$ to each $m \in \mathbb{M}$;
- $I : \mathbb{P} \to \wp(W)$ associates an interpretation $I(p)$ to each $p \in \mathbb{P}$.

The set of all Kripke models is noted $\mathcal{K}$.

The size of a Kripke model $M = \langle W, R, I\rangle$, which is the number of possible worlds in W, is noted $|M|$.

For some applications, it is more convenient to define valuations instead of interpretations. Kripke models defined in such way are triplets of the form $\langle W, R, V\rangle$, where W and R are as in Definition 3.4 and we have:

- $V : W \to (\mathbb{P} \to \{0, 1\})$ associates a Boolean valuation to each $w \in W$.

Note that both definitions are equivalent. One can go from one to the other by letting:

$$V(w)(p) = \begin{cases} 1, \text{if } w \in I(p)\\ 0, \text{otherwise}\end{cases}$$

To simplify notation, we sometimes use R_m instead of $R(m)$ and, analogously, I_p instead of $I(p)$ and V_w instead of $V(w)$. We also sometimes use $R_m(w)$ to denote the set $\{w' \mid (w, w') \in R(m)\}$.

Definition 3.5 (Pointed Kripke Model)**.** A *pointed Kripke model* is a pair $\langle\langle W, R, I\rangle, w\rangle$, where $\langle W, R, I\rangle$ is a Kripke model and w is a distinguished possible world from W called *actual world* or *root* of the model.

$$
\begin{aligned}
\mathbb{P} &= \{p_0, p_1\} \\
\mathbb{M} &= \{m_0\} \\
W &= \{w_0, w_1, w_2\} \\
R_{m_0} &= \{(w_0, w_1), (w_0, w_2), (w_1, w_1)\} \\
V_{w_0} &= \emptyset \\
V_{w_1} &= \{p_0\} \\
V_{w_2} &= \{p_1\}
\end{aligned}
$$

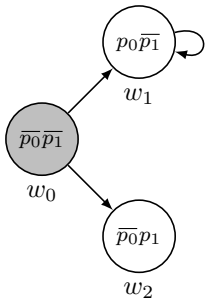


Figure 3.1 – A Kripke model $\langle M, w_0 \rangle$

For convenience, sometimes we just drop one '$\langle$', and use $\langle W, R, V, w \rangle$ to denote a pointed Kripke model. Also for convenience, we often use $\langle M, w \rangle$, where $M = \langle W, R, I \rangle$, to denote a pointed Kripke model.

Pointed Kripke models can be seen as a particular kind of graph. An example of a pointed Kripke model is given in Figure 3.1. We often use such pictures to show Kripke models in this thesis.

Definition 3.6 (Satisfaction Relation)**.** The satisfaction relation $\vDash$ between formulas in $\mathcal{L}_{\mathrm{ML}}$ and pointed Kripke models is recursively defined as follows:

$$
\begin{aligned}
&\langle M, w \rangle \vDash \top \\
&\langle M, w \rangle \vDash p && \text{iff} && w \in I_p \\
&\langle M, w \rangle \vDash \neg\varphi && \text{iff} && \langle M, w \rangle \nvDash \varphi \\
&\langle M, w \rangle \vDash \varphi \wedge \psi && \text{iff} && \langle M, w \rangle \vDash \varphi \text{ and } \langle M, w \rangle \vDash \psi \\
&\langle M, w \rangle \vDash \varphi \vee \psi && \text{iff} && \langle M, w \rangle \vDash \varphi \text{ or } \langle M, w \rangle \vDash \psi, \text{ or both} \\
&\langle M, w \rangle \vDash [m]\varphi && \text{iff} && \text{for all } w' \in R_m(w) \text{ we have } \langle M, w' \rangle \vDash \varphi \\
&\langle M, w \rangle \vDash \langle m \rangle\varphi && \text{iff} && \text{there is } w' \in R_m(w) \text{ such that } \langle M, w' \rangle \vDash \varphi
\end{aligned}
$$

For example, let M be the Kripke model in Figure 3.1, we have:

$$
\begin{aligned}
&\langle M, w \rangle \nvDash p_0 && \langle M, w \rangle \vDash \neg p_1 \\
&\langle M, w \rangle \vDash \langle m_0 \rangle p_0 && \langle M, w \rangle \nvDash [m_0] p_0 \\
&\langle M, w \rangle \nvDash \langle m_0 \rangle (p_0 \wedge p_1) && \langle M, w \rangle \vDash [m_0](p_0 \vee p_1) \\
&\langle M, w \rangle \vDash \langle m_0 \rangle [m_0] p_0 && \langle M, w \rangle \nvDash [m_0]\langle m_0 \rangle p_0
\end{aligned}
$$

Let $\varphi \in \mathcal{L}_{\mathrm{ML}}$. We sometimes need to refer to the *extension* of φ in the Kripke model $M = \langle W, R, I \rangle$. This is the set $[\![\varphi]\!]_M = \{w \mid \langle M, w \rangle \vDash \varphi\}$.

Definition 3.7 (Validity)**.** A formula $\varphi \in \mathcal{L}_{\mathrm{ML}}$ is *valid in a Kripke model* $M = \langle W, R, I \rangle$ (noted $M \vDash \varphi$) if and only if, for all possible worlds $w \in W$, we have $\langle M, w \rangle \vDash \varphi$. A formula $\varphi \in \mathcal{L}_{\mathrm{ML}}$ is *valid* (noted $\vDash \varphi$) if and only if, for all Kripke models $M \in \mathcal{K}$, we have $M \vDash \varphi$.

(CPL)	All axioms schemas in Table 2.1	
(Df$\Diamond$)	$\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$	(definition of $\Diamond$)
(K)	$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$	(distributivity)
(RMP)	From φ and $\varphi \to \psi$ infer ψ	(modus ponens)
(RN)	From ψ infer $\Box\psi$	(necessitation)

Table 3.1 – Axiom system of modal logic K

Definition 3.8 (Satisfiability)**.** A formula $\varphi \in \mathcal{L}_{\mathrm{ML}}$ is *satisfiable* if and only if $\not\models \neg\varphi$.

Corollary 3.1. *A formula φ is satisfiable if and only if there is a Kripke model $M \in \mathcal{K}$ and a possible world $w \in W$ such that $\langle M, w\rangle \vDash \varphi$.*

When a pointed model $\langle M, w\rangle$ satisfies a formula $\varphi \in \mathcal{L}_{\mathrm{ML}}$, we sometimes say that the pointed model is *a model of* φ.

Definition 3.9 (Semantic Consequence)**.** Let two formulas $\varphi, \psi \in \mathcal{L}_{\mathrm{ML}}$ be given. We say that ψ *is a consequence of* φ, noted $\varphi \vDash \psi$, if and only if, for all $\langle M, w\rangle$ such that $\langle M, w\rangle \vDash \varphi$ we have $\langle M, w\rangle \vDash \psi$.

3.4 Axiom Systems of Modal Logic

There are several different modal logics. The simplest and one of the most common modal logic in the literature is called K. Its language is $\mathcal{L}_{\mathrm{ML}}$, where the vocabulary contains only one modal context. Its axiom system is the one in Table 3.1.

The presence of principles (CPL) and (RMP) in the axiom system of K means that it is a conservative extension of CPL. In other words, the axiom system of K proves all CPL tautologies, including the additional tautologies that can be written using the operators $\Box$ and $\Diamond$. For example, $\Box p \vee \neg\Box p$ is valid in K as well as $p \vee \neg p$ in CPL. The principle (Df$\Diamond$) stipulates the duality between operators $\Box$ and $\Diamond$. It defines operator $\Diamond$ in terms of operator $\Box$. This is the formal counterpart of the equivalence between formulas (3.1) and (3.2) seen before. This is also why, sometimes, the operator $\Diamond$ is defined as an abbreviation in modal logic (similarly to what we could have done with the connective $\vee$). Principle (K) is a kind of deductive closure for $\Box$. It stipulates that, if φ necessarily implies ψ, then necessarily φ implies necessarily ψ. Principle (RN) stipulates that all valid formulas is necessarily true. This is intuitive, since necessarily true means true in all possible worlds.

As an example, let us see a derivation in the axiom system of K.

1.	$(\varphi_1 \wedge \cdots \wedge \varphi_n) \to \psi$	(hypothesis)
2.	$\varphi_1 \to (\cdots \to (\varphi_n \to \psi))$	(from 1 with CPL, RMP)
3.	$\Box(\varphi_1 \to (\cdots \to (\varphi_n \to \psi)))$	(from 2 with RN)
4.	$\Box\varphi_1 \to \Box(\cdots \to (\varphi_n \to \psi))$	(from 3 with K, CPL and RMP)
	$\vdots$	
$n+4$.	$\Box\varphi_1 \to (\cdots \to (\Box\varphi_n \to \Box\psi))$	(from $n+3$ with K, CPL and RMP)
$n+5$.	$(\Box\varphi_1 \wedge \cdots \wedge \Box\varphi_n) \to \Box\psi$	(from $n+4$ with K, CPL and RMP)

The latter derivation shows that the rule (RK) below is is a valid infernece rule in K.

(RK) From $(\varphi_1 \wedge \cdots \wedge \varphi_n) \to \psi$ (general modal consequence)
infer $(\Box\varphi_1 \wedge \cdots \wedge \Box\varphi_n) \to \Box\psi$

Indeed, there is an alternative axiom system for K consisting of axioms (CPL), (Df$\Diamond$), and the inference rules (RMP) (RK).

The reader may verify that (CPL) and axiom schemas (Df$\Diamond$) and (K) are valid in logic K. In fact, Kripke models satisfy precisely the valid formulas in K or, in other words, the set of validities is determined by the class of models $\mathcal{K}$, as shown below.

Theorem 3.2 (Soundness and Completeness of K)**.** $\vdash \varphi$ *if and only if* $\vDash \varphi$.

The implication from the left to the right is rather easy. It is proved by showing that all models in $\mathcal{K}$ satisfy all principles of K. In other words, we show that each axiom is valid in $\mathcal{K}$ and, for each inference rule, if its antecedent is valid in $\mathcal{K}$ so is its consequent.

The other direction is proved in four steps:

1. We define the canonical model $M^c = \langle W^c, R^c, I^c\rangle$ such that:
   - W^c is the set of all maximally consistent sets in K;
   - For all $w, w' \in W, (w, w') \in R^c_m$ if and only if $\{\varphi \mid [m]\varphi \in w\} \subseteq w'$; and
   - $I^c_p = \{w \mid p \in w\}$
2. We prove Lemma 1: The canonical model M^c is a Kripke model.
3. We prove the Truth Lemma: $M^c \vDash \varphi$ if and only if $\varphi \in$ K.
4. We have: if $\vDash \varphi$ then $M^c \vDash \varphi$, by Lemma 1. Then $\varphi \in$ K, by the Truth Lemma.

We also note that modal logic K is *strongly complete*. This means the following.

Theorem 3.3. $\varphi \vDash \psi$ *if and only if* $\varphi \vdash \psi$.

3.4.1 Other Modal Logics

Different modal logics can be defined using the axiom system of K plus one or more axiom schemas. The most common axiom schemas found in literature are the following:

(D) $\Box\varphi \to \Diamond\varphi$
(T) $\Box\varphi \to \varphi$
(B) $\varphi \to \Box\Diamond\varphi$
(4) $\Box\varphi \to \Box\Box\varphi$
(5) $\Diamond\varphi \to \Box\Diamond\varphi$

For example, the modal logic KT is the modal logic containing all principles of K plus the axiom schema T; the modal logic KD5 is K plus axiom schemas D and 5; the modal logic KD45 is K plus axiom schemas D, 4 and 5; etc.. More formally, we have the following.

Definition 3.10 (Axiom System of Logic KX)**.** Let $X \subseteq \{\mathrm{D}, \mathrm{T}, \mathrm{B}, 4, 5\}$. The axiom system of modal logic KX consists of axiom schemas (CPL), (Df$\Diamond$), (K) plus the ones in X, and the inference rules (RMP) and (RN).

It is interesting to note that, for instance, the set of validities of KT contains the validities of K. This is so because all formulas that can be derived using the axiom system of K can also be derived using the axiom system of KT. The same holds for all modal logics KX. It is even more interesting to note that some axiom schemas imply others. For example, axiom schema T implies axiom schema D. This means that KT contains KD. As another example, we have that axiom schemas K T and 5 together imply all the other ones, namely, B, D and 4. In fact, all combinations of these schemas give rise to 15 different axiom systems. Figure 3.2 presents the inclusions between them. Proofs for all these results can be found, for instance, in (Chellas 1980).

Soundness and completeness for these logics are proved as for Theorem 3.2, but the set of Kripke models is different for each one of them. We delay this until Section 3.5, when we see the list of properties that correspond to the axiom schemas considered here.

3.5 Expressiveness

It turns out that modal logic is a fragment of first-order logic (FOL). Below, we see how to compute an equisatisfiable FOL formula from a ML formula.

Definition 3.11 (Standard Translation)**.** Assume:

- a denumerable set of variables $\{x, y, \ldots\}$,
- an unary predicate p for each propositional variable $p \in \mathbb{P}$, and
- a binary predicate R_m for each modality in $\mathbb{M}$.

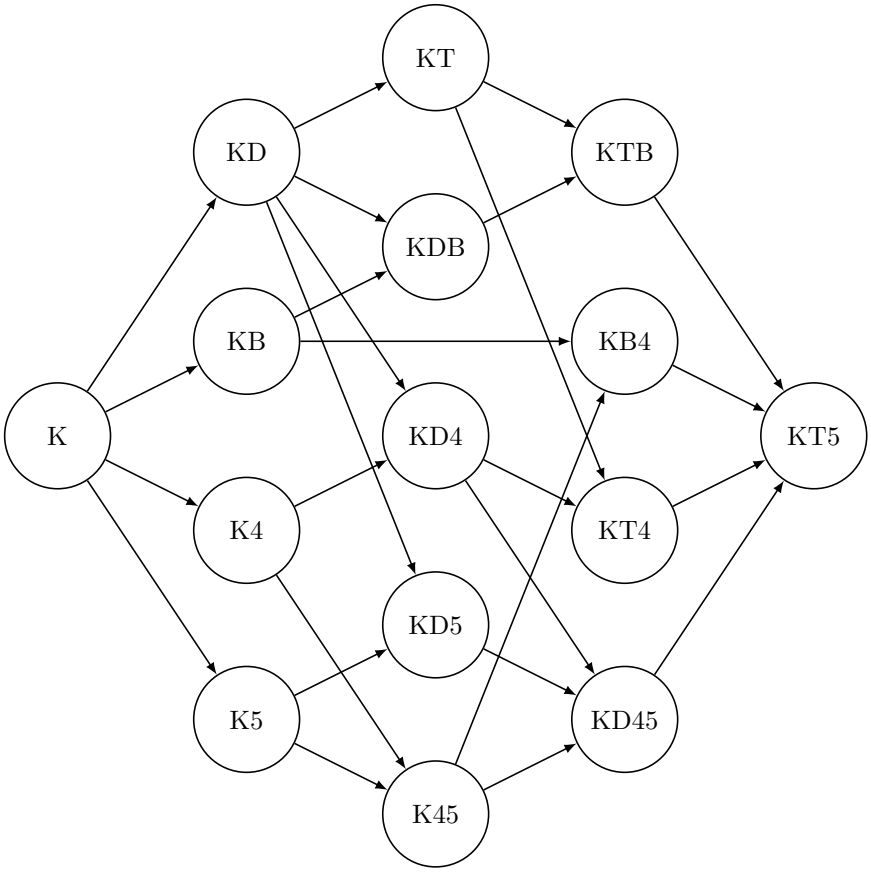


Figure 3.2 – Modal logic axiom systems inclusions. A path from L_1 to L_2 means that all validities in L_1 are also in L_2.

$$
\begin{aligned}
&\mathrm{str}(\bot, x) = \bot \\
&\mathrm{str}(p, x) = p(x) \\
&\mathrm{str}(\neg\varphi, x) = \neg\,\mathrm{str}(\varphi, x) \\
&\mathrm{str}(\varphi \to \psi, x) = \mathrm{str}(\varphi, x) \to \mathrm{str}(\psi, x) \\
&\mathrm{str}(\varphi \wedge \psi, x) = \mathrm{str}(\varphi, x) \wedge \mathrm{str}(\psi, x) \\
&\mathrm{str}(\varphi \vee \psi, x) = \mathrm{str}(\varphi, x) \wedge \mathrm{str}(\psi, x) \\
&\mathrm{str}([m]\varphi, x) = \forall y(R_m(x, y) \to \mathrm{str}(\varphi, y)) \\
&\mathrm{str}(\langle m\rangle\varphi, x) = \exists y(R_m(x, y) \wedge \mathrm{str}(\varphi, y)) \qquad \text{(for a fresh } y\text{)}
\end{aligned}
$$

Theorem 3.4. *For any formula φ, any Kripke model M, and any world w in M, we have that $\langle M, w\rangle \vDash \varphi$ if and only if $M \vDash_{\mathrm{FOL}} \mathrm{str}(\varphi, x)[x \leftarrow w]$.*[4]

Example 3.1. Just to be sure that we understood the principle, let us see a small example of translation:

$$
\begin{aligned}
\mathrm{str}(p_0 \wedge [m_0]p_0) &= \mathrm{str}(p_0 \wedge [m_0]p_0, x) \\
&= \mathrm{str}(p_0, x) \wedge \mathrm{str}([m_0]p_0, x) \\
&= p_0(x) \wedge \forall y(R_{m_0}(x, y) \to \mathrm{str}(p_0, y)) \\
&= p_0(x) \wedge \forall y(R_{m_0}(x, y) \to p_0(y))
\end{aligned}
$$

■

This result is important for us. We will see, on Chapter 7, that we can speed up reasoning on modal logic by adapting this translation to a translation to CPL, and then giving the resulting formula to a SAT solver.

This result is also very important for the field of modal logics itself. This is how modal logics have "inherited for free" various properties of FOL, such as compactness and semi-decidability. But some FOL properties are not the same in ML. One of the most important is called *invariance*.

Theorem 3.5. *If f is an isomorphism between first-order logic models M and M' then for each first-order formula $\varphi(x_1, \ldots, x_k)$, and each matching tuple of objects $\langle o_1, \ldots, o_k\rangle$ in M we have that:*

$$M \vDash \varphi[o_1, \ldots o_k] \quad \textit{if and only if} \quad M' \vDash \varphi[f(o_1), \ldots f(o_k)]$$

The converse is also true for finite models.

In other words, two FOL models satisfy the same set of formulas if and only if they are isomorphic. It is a curious fact that this does not transfer to modal logic. To see it, first consider the definition of isomorphic Kripke models.

Definition 3.12 (Isomorphism)**.** Let $M = \langle W, R, V\rangle$ and $M' = \langle W', R', V'\rangle$ be two Kripke models. An *isomorphism* between M and M' is a bijection $f : W \to W'$ such that:

[4] Proofs or proof sketches of theorems 3.4, 3.5, 3.6, 3.7, 3.8 can be found in (Blackburn et al. 2007).

Figure 3.3 – Equivalent, but not isomorphic, Kripke models

- $(w, v) \in R$ iff $(f(w), f(v)) \in R$;
- $w \in V(p)$ iff $f(w) \in V'(p)$.

Theorem 3.6. *Let f be an isomorphism between two Kripke models M and M'. For all formulas $\varphi \in \mathcal{L}_{\mathrm{ML}}$ and all worlds $w \in W$, we have that $\langle M, w\rangle \vDash \varphi$ if and only if $\langle M', f(w)\rangle \vDash \varphi$.*

However, the converse of Theorem 3.6 is not true. We can see that by analysing the two models in Figure 3.3. It is not possible to find a modal logic formula that is true in one of them and false in the other one. Nonetheless, they are clearly not isomorphic.

Therefore, modal logics do not distinguish all non-isomorphic models. This means that they are strictly less expressive than FOL. In fact, modal logic invariance comes in a different flavor than in FOL. Actually, ML cannot distinguish *bisimilar* models. Let us see the formal definition.

Definition 3.13 (Bisimulation)**.** Let $M = \langle W, R, V\rangle$ and $M' = \langle W', R', V'\rangle$ be two Kripke models. A *bisimulation* between M and M' is a binary relation $B \subseteq W \times W'$ such that for all $(w_0, w'_0) \in B$ we have:

Atomic harmony: $\langle M, w_0\rangle \vDash p$ if and only if $\langle M', w'_0\rangle \vDash p$;

Zig: if $(w_0, w_1) \in R_m$ then there exists $w'_1 \in W'$ such that $(w_1, w'_1) \in B$ and $(w'_0, w'_1) \in R'_m$;

Zag: if $(w'_0, w'_1) \in R'_m$ then there exists $w_1 \in W$ such that $(w_1, w'_1) \in B$ and $(w_0, w_1) \in R_m$.

Definition 3.14 (Bisimilarity)**.** Two pointed Kripke models are *bisimilar*, which is noted $\langle M, w\rangle \leftrightarroweq \langle M', w'\rangle$, if and only if there is a bisimulation B between M and M' such that $(w, w') \in B$.

The two Kripke models in Figure 3.3 are examples of bisimilar Kripke models that are not isomorphic. The reader may verify that the relation $B = \{(w_0, w'_0), (w_0, w'_1)\}$ is a bisimulation between M and M'.

Theorem 3.7. *If two pointed Kripke models $\langle M, w\rangle$ and $\langle M', w'\rangle$ are bisimlar then $\langle M, w\rangle$ and $\langle M', w'\rangle$ satisfy the same formulas in $\mathcal{L}_{\mathrm{ML}}$.*

Theorem 3.8. *If M and M' are finite Kripke models and $\langle M, w\rangle$ and $\langle M', w'\rangle$ satisfy the same set of formulas then they are bisimilar.*

Bisimulation is therefore a key concept in modal logics. We use it a couple of times in this thesis, whenever we need to show that two Kripke models are equivalent. This happens, for instance, in Chapter 6 where we need to show that two different ways of revising an epistemic model result in the same (up to bisimulation) model.

Bisimulation also plays an important role on the comprehension of another key concept in modal logics called *correspondence theory* (van Benthem 1984). Recall that we saw the definition of several different logics in Section 3.4.1. Each one of those logics "corresponds" to a particular class of Kripke models. This result is captured by the next theorem.

Theorem 3.9 (First-order Definability (Lemmon and Scott 1977))**.** *Let the logic* L *be defined by* (CPL)*,* (Df$\Diamond$)*,* (K)*,* (RMP)*,* (RN) *and a set A of axiom schemas of the form:*

$$(\mathrm{G}^{i,j,k,\ell}) \qquad \langle m\rangle^i[m]^j\varphi \to [m]^k\langle m\rangle^\ell\varphi \qquad (i,j,k,\ell \geq 0)$$

L *is determined by the class of Kripke models satisfying a condition:*

$$(\mathrm{C}^{i,j,k,\ell}) \qquad \forall x\forall y\forall z((R^i_m(x,y) \wedge R^k_m(x,z)) \to \exists v(R^j_m(y,v) \wedge R^\ell_m(z,v)))$$

for each axiom schema in A.

Corollary 3.10.

1. *Axiom schema* (D) $= \mathrm{G}^{0,1,0,1} = [m]\varphi \to \langle m\rangle\varphi$ *corresponds to seriality, i.e.:*[5]

$$\forall x\forall y\forall z((R^0_m(x,y) \wedge R^0_m(x,z)) \to \exists v(R^1_m(y,v) \wedge R^1_m(z,v))) \quad \equiv \quad \forall y\exists v(R_m(y,v))$$

2. *Axiom schema* (T) $= \mathrm{G}^{0,1,0,0} = [m]\varphi \to \varphi$ *corresponds to reflexivity, i.e.:*

$$\forall x\forall y\forall z((R^0_m(x,y) \wedge R^0_m(x,z)) \to \exists v(R^1_m(y,v) \wedge R^0_m(z,v))) \quad \equiv \quad \forall x(R_m(x,x))$$

3. *Axiom schema* (B) $= \mathrm{G}^{0,0,1,1} = \varphi \to [m]\langle m\rangle\varphi$ *corresponds to symmetry, i.e.:*

$$\forall x\forall y\forall z((R^0_m(x,y) \wedge R^1_m(x,z)) \to \exists v(R^0_m(y,v) \wedge R^1_m(z,v))) \quad \equiv \quad \forall x\forall z(R_m(x,z) \to (R_m(z,x)))$$

4. *Axiom schema* (4) $= \mathrm{G}^{0,1,2,0} = [m]\varphi \to [m][m]\varphi$ *corresponds to transitivity, i.e.:*

$$\forall x\forall y\forall z((R^0_m(x,y) \wedge R^2_m(x,z)) \to \exists v(R^1_m(y,v) \wedge R^0_m(z,v))) \quad \equiv \quad \forall x\forall z(R^2_m(x,z) \to R_m(x,z))$$

5. *Axiom schema* (5) $= \mathrm{G}^{1,0,1,1} = \langle m\rangle\varphi \to [m]\langle m\rangle\varphi$ *corresponds to Euclideanity:*

$$\forall x\forall y\forall z((R^1_m(x,y) \wedge R^1_m(x,z)) \to \exists v(R^0_m(y,v) \wedge R^1_m(z,v))) \quad \equiv \quad \forall x\forall y\forall z((R_m(x,y) \wedge R_m(x,z)) \to R_m(z,y))$$

[5] Note that $R^0_m(x,y)$ means $x = y$.

Schema	First-Order Property
(K) $[m](\varphi \to \psi) \to ([m]\varphi \to [m]\psi)$	None
(T) $[m]\varphi \to \varphi$	Reflexivity
(B) $\varphi \to [m]\langle m\rangle\varphi$	Symmetry
(D) $[m]\varphi \to \langle m\rangle\varphi$	Seriality
(4) $[m]\varphi \to [m][m]\varphi$	Transitivity
(5) $\langle m\rangle\varphi \to [m]\langle m\rangle\varphi$	Euclideanity

Table 3.2 – Axiom schemas and their corresponding structural properties

We summarise all results from Corollary 3.10 in Table 3.2.
Lemmon and Scott's result has been generalised further by Sahlqvist (1975).

Definition 3.15 (Salhqvist Formula)**.** A *Sahlqvist formula* is a formula of the form

$$\langle m\rangle^n(\varphi \to \psi)$$

where $n \geq 0$ and φ and ψ satisfy the following conditions:

1. no operators occur in φ except $\langle m\rangle$, $[m]$, $\vee$, $\wedge$ and $\neg$;
2. operator $\neg$ occurs only immediately before a variable in φ;
3. no occurrence of $[m]$, $\vee$ or $\wedge$ lies within the scope of any $\langle m\rangle$ in φ;
4. no operators occur in φ except $\langle m\rangle$, $[m]$, $\vee$ and $\wedge$ ($\neg$ is not permitted).

Theorem 3.11 (Sahlqvist Theorem (Sahlqvist 1975))**.** *There is an effective method for computing first-order equivalents for Sahlqvist formulas.*

3.6 Computational Complexity

It is a known fact that model checking in normal modal logics can be computed in $O(n)$, where n is the length of the formula given as input. The algorithm performing such a task (Algorithm 3.1) is the most obvious implementation of Definition 3.6. Note, however, that this assumes an explicit representation of the Kripke model given as input. If a more clever representation of the Kripke model is used, the complexity of model checking may be higher. But this task is never more complex than satisfiability checking, which has been proved to be PSACE-Complete for most modal logics and NP-Complete in the case of KT5.

PSPACE-Hardness for satisfiability checking in K, KT and KT4, as well as NP-Completeness for KT5 has been proved in a famous paper by Ladner (1977). In another

input: A pointed Kripke model $\langle M, w\rangle$ and a formula $\varphi \in \mathcal{L}_{\mathrm{ML}}$
output: **true** if $\langle M, w\rangle \vDash \varphi$, **false** otherwise

```
1  function mlmc(⟨M, w⟩, φ)
2      if φ = ⊤ then
3          return true
4      if φ ∈ ℙ then
5          if w ∈ I(φ) then
6              return true
7          else
8              return false
9      if φ = ψ0 ∧ ψ1 then
10         return mlmc(⟨M, w⟩, ψ0) and mlmc(⟨M, w⟩, ψ1)
11     if φ = ψ0 ∨ ψ1 then
12         return mlmc(⟨M, w⟩, ψ0) or mlmc(⟨M, w⟩, ψ1)
13     if φ = [m]ψ then
14         forall w' ∈ R_{m0}(w) do
15             if not mlmc(⟨M, w⟩, ψ) then
16                 return false
17         return true
18     if φ = ⟨m⟩ψ then
19         forall w' ∈ R_{m0}(w) do
20             if mlmc(⟨M, w⟩, ψ) then
21                 return true
22         return false
```

Algorithm 3.1: Modal Logic Model Checking

famous paper, Halpern and Moses (1992) extended those results for the multi-modal versions of those logics plus KD45 and also studied the addition of distributed and common knowledge operators. NP-Completeness of satisfiability checking has been proved for all mono-modal logics with axiom (5) by Halpern and Rêgo (2007a,b). Here, we see a brief summary of the results that are interesting for us in this thesis.

The proof sketch of the next theorem gives us a fairly easy way to show that all modal logics we have seen in this chapter are decidable. The technique used is called filtration. It mainly shows that there is a finite model for every satisfiable formula, and also presents the maximum size of such a model. We also use this technique to prove decidability of the logics presented in chapters 4 and 5.

Theorem 3.12. *If φ is satisfiable, then there is a Kripke model containing at most $2^{|\operatorname{sub}(\varphi)|}$ possible worlds satisfying it, where $\operatorname{sub}(\varphi)$ is the set of sub-formulas of φ.*

Proof Sketch. Let $M = \langle W, R, I\rangle$ be a Kripke model satisfying φ. First, we build equivalence classes of possible worlds. Two possible worlds w and w' are in the same class (noted $w \equiv w'$) if and only if:

$$\text{for all } \psi \in \operatorname{sub}(\varphi) \text{ we have that } \langle M, w\rangle \vDash \psi \text{ iff } \langle M, w'\rangle \vDash \psi$$

Now, let the equivalence class of a possible world w be $[w] = \{w' \mid w \equiv w'\}$. The filtration of M by the formula φ is a Kripke model $M^f = \langle W^f, R^f, I^f\rangle$, where:

- $W^f = \{[w] \mid w \in W\}$
- $R^f([w_0], [w_1])$ iff there exists $w'_0 \in [w]$, and $w'_1 \in [w_1]$ such that $R(w'_0, w'_1)$
- $I^f(p) = \{[w] \mid \langle M, w\rangle \vDash p\}$

Note that there cannot be more than $2^{|\operatorname{sub}(\varphi)|}$ worlds in M^f, because this is the number of ways possible worlds can disagree, with respect to φ.

The next thing we must do is to show that, for all $\psi \in \operatorname{sub}(\varphi)$, $\langle M, w\rangle \vDash \psi$ iff $\langle M^f, [w]\rangle \vDash \psi$. This is be done by an easy induction on the structure of ψ.

In the induction base, we have $\psi = p$ for some $p \in \mathbb{P}$. $\langle M, w\rangle \vDash p$ iff $[w] \in I^f(p)$ (by the definition of I^f) iff $\langle M^f, [w]\rangle \vDash p$.

There are five cases on the induction step, one for each Boolean connective and modal operator in the language. The cases for the Boolean connectives are straightforward. For the modal operator $\Box$, let $\psi = \Box\chi$. $\langle M, w\rangle \vDash \Box\chi$ iff, for all $w' \in R(w)$, we have $\langle M, w'\rangle \vDash \chi$. The latter is true iff, for all $w'' \in [w']$, we have $\langle M, w''\rangle \vDash \chi$ (by the definition of $[w']$). We also have that $([w], [w']) \in R^f$ (by the definition of R^f). Therefore, we have that the former is true iff, for all $[w'] \in R([w])$ we have $\langle M^f, [w']\rangle \vDash \chi$, iff $\langle M^f, [w]\rangle \vDash \Box\chi$. The induction step for the modal operator $\Diamond$ is analogous. ∎

From Theorem 3.12, we obtain the so-called *finite model property* for the modal logics we have seen here. This, together with some properties of their axiom system, gives us decidability of satisfiability checking.

Theorem 3.13. *Satisfibility checking in all modal logics of Figure 3.2 are decidable.*

	$n = 1$	$n > 1$
$\mathrm{K_n}$ $\mathrm{KT_n}$ $\mathrm{KT4_n}$	PSPACE-Complete	PSPACE-Complete
$\mathrm{K} * 5_\mathrm{n}$	NP-Complete	PSPACE-Complete

Table 3.3 – Computational complexites of satisfiability checking

Proof Sketch. By Definition 2.8 and Theorem 3.2, there is a finite derivation for each valid formula. Also, the set of axiom schemas and inference rules is finite. This means that the set of derivations is enumerable. Therefore, if $\neg\varphi \in \mathcal{L}_{\mathrm{ML}}$ is valid (i.e., if it is not satisfiable), one can find its derivation in such enumeration. On the other hand, by Theorem 3.12, there is a finite model satisfying every satisfiable formula. Because the set of such models is also enumerable, if φ is satisfiable, one can find the model satisfying φ in such enumeration. Therefore, there is an algorithm that decides whether φ is satisfiable, which is as follows: It generates the next derivation and verify if it proves that $\neg\varphi$ is valid. If so, it returns false. If not, it generates the next Kripke model and verify if it satisfies φ. If so, it return true. If not, it loops. Eventually, the algorithm stops either with a proof that $\neg\varphi$ is valid or with a Kripke model for φ. ∎

The procedure described in the proof sketch of Theorem 3.13 is far from optimal. Optmial procedures for deciding satisfiability of modal formulas can be found, for instance, in (Goré 1999). The methods proposed therein are tableau methods. Their optimality rely in the following results.

Lemma 3.14. *Let $M = \langle W, R, I\rangle$ be a Kripke model, where $w \in W$. Let $n \in \mathbb{N}$ and let $M|_{w,n}$ be the Kripke model $\langle W|_{w,n}, R|_{w,n}, I|_{w,n}\rangle$ obtained from M such that:*

- $W|_{w,n} = \{w\} \cup \{w' \mid R^{\leq n}(w, w')\}$
- $R|_{w,n} = R \cap W|_{w,n}$
- $I|_{w,n} = I \cap W|_{w,n}$

Then, for all $\varphi \in \mathcal{L}_{\mathrm{ML}}$, if $\mathrm{md}(\varphi) \leq n$ then, $\langle M, w\rangle \vDash \varphi$ if and only if $\langle M|_{w,n}, w\rangle \vDash \varphi$.

The tableau methods for modal logics expoits Lemma 3.14 and also the fact that $|\,\mathrm{sub}(\varphi)| \leq \mathrm{len}(\varphi)$. They explore the Kripke model for φ branch by branch, without having to put it entirely on the memory. The result is a method that is exponential in time but polynomial is space.

Theorem 3.15. *Every formula φ in $\mathcal{L}$ is satisfiable in a model based on a finite tree of depth at most $\mathrm{md}(\varphi)$.*

Corollary 3.16. *Satisfiability checking in* K *is in PSPACE.*

The result above, with some clever adjustments, can be extended to all modal logics we have seen. We summarize computational complexity in Table 3.3.

In Section 7.2, we use a tableau to show the adequacy of a new method for satisfiability checking in KT5. We therefore take a look on tableau methods for modal logic

here. First, we need to define a tableau for a modal logic formula. The definition below defines tableau for the modal logic K. It is an extension of the tableau for a CPL formula that we saw in Definition 2.14.

Definition 3.16 (ML Tableau)**.** Let $\varphi \in \mathcal{L}_{\text{ML}}$. A *tableau for* φ is a set T of pairs of the form (σ, φ). The first element of the pair is called *label*. It is a (possibly empty) sequence of natural numbers and modalities from $\mathbb{M}$. The second element of the pair is a formula $\psi \in \text{sub}(\varphi)$. In addition, we have that $(0, \varphi) \in T$ and, for all sequences σ, T satisfies the following conditions:

1. $(\sigma, \neg\top) \notin T$.
2. $(\sigma, \psi) \in T$ if and only if $(\sigma\neg\psi) \notin T$.
3. if $(\sigma, \neg\neg\psi) \in T$ then $(\sigma, \psi) \in T$.
4. if $(\sigma, \psi_1 \wedge \psi_2) \in T$ then $(\sigma, \psi_1) \in T$ and $(\sigma, \psi_2) \in T$.
5. if $(\sigma, \psi_1 \vee \psi_2) \in T$ then $(\sigma, \psi_1) \in T$ or $(\sigma, \psi_2) \in T$.
6. if $(\sigma, [m]\psi) \in T$ then, for all $(\sigma mi, \chi) \in T$, we have $(\sigma mi, \psi) \in T$.
7. if $(\sigma, \langle m\rangle\psi) \in T$ then $(\sigma mi, \psi) \in T$, for some $i \in \mathbb{N}$.

The tableau method is presented in Algorithm 3.2. As for CPL, it starts with the singleton $T = \{(0, \varphi)\}$ and decomposes φ by adding its sub-formulas to T, aiming at satisfying all conditions in Definition 3.16. Here though, it also handles a set of labels σ, which represent the possible worlds in the model being constructed for φ. For example, if the pair $(\sigma, \langle m\rangle\varphi)$ is in T, then the method choses a fresh (not yet present in T) natural number i and adds $(\sigma mi, \varphi)$ to T. This means that, whenever $\langle m\rangle\varphi$ is true in the possible world σ then there exists a possible world σmi where ψ is true. A dual condition handles the operator $[m]$. The initial sequence 0 represents the actual world in the model.

As for the CPL tableau method, for the branches where Algorithm 3.2 returns **true**, one can construt a Kripke model M, where:

- $W = \{i \mid (\sigma i, \chi) \in T\}$, for some χ.
- $R_m = \{(i, j) \mid (\sigma imj, \chi) \in T\}$, for some χ.
- $V(p) = \{i \mid (\sigma i, p) \in T\}$

An easy induction on the structure of φ shows that $\langle M, 0\rangle \vDash \varphi$.

An exemple of the execution of Algorthm 3.2 in a modal logic K with only one modality is shown in Figure 3.4. The closed left branch is discarded, but a model $\langle M, 0\rangle$ satisfaying φ can be build from the right branch of the tree. We have $M = \langle W, R, V\rangle$, where: $W = \{0, 1\}$, $R_{m_0} = \{(0, 1)\}$, $V_p = \{1\}$ and $V_q = \{1\}$.

Additional conditions must be added to Definition 3.16 for the other modal logics in Figure 3.2. For instance, modal logic KT4 has the following conditions added:

input: The set $T = \{\varphi\}$
output: **true** if φ is satisfiable, **false** otherwise

```
1  function mltableau(T)
2      if (σ, ⊥) ∈ T or {(σ, φ), (σ, ¬φ)} ⊆ T then
3          return false
4      if (σ, ¬¬ψ) ∈ T then
5          return mltableau(T \ {(σ, ¬¬ψ)} ∪ {(σ, ψ)})
6      if (σ, ψ1 ∧ ψ2) ∈ T then
7          return mltableau(T \ {(σ, ψ1 ∧ ψ2)} ∪ {(σ, ψ1), (σ, ψ2)})
8      if (σ, ψ1 ∨ ψ2) ∈ T then
9          return mltableau(T \ {(σ, ψ1 ∨ ψ2)} ∪ {(σ, ψ1)}) or
             mltableau(T \ {(σ, ψ1 ∨ ψ2)} ∪ {(σ, ψ2)})
10     if {(σ, [m]ψ), (σmi, χ)} ∈ T and (σmi, ψ) ∉ T then
11         return mltableau(T ∪ {(σmi, ψ)})
12     if (σ, ⟨m⟩ψ) ∈ T then
13         return mltableau(T \ {(σ, ⟨m⟩ψ)} ∪ {(σmi, ψ)})
```

Algorithm 3.2: Tableau Method for Modal Logic

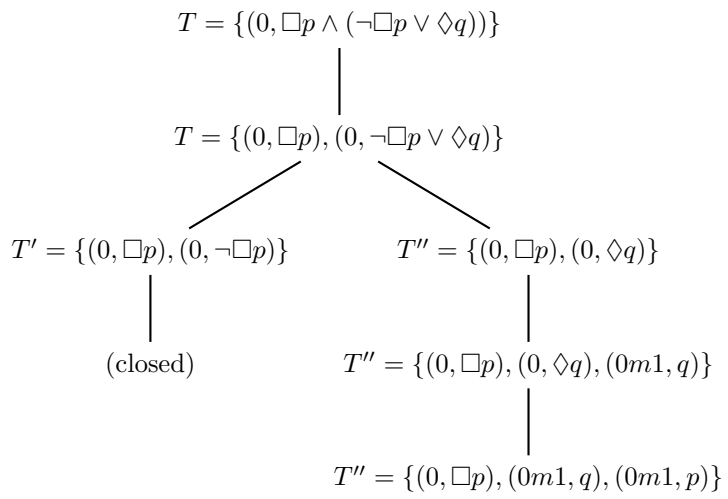


Figure 3.4 – Execution of the tableau method for $\Box p \wedge (\neg\Box p \vee \Diamond q)$

- If $(\sigma, [m]\varphi) \in T$ then $(\sigma, \varphi) \in T$
- If $(\sigma, [m]\varphi) \in T$ then $(\sigma, [m][m]\varphi) \in T$

The first condition corresponds to the addition of axiom schema (T) to the logic, whereas the second condition corresponds to axiom schema (4). The corresponding tableau method must also handle these two additional conditions.[6] The reader can consult (Goré 1999) for more details. For the logic KT5, the method can actually be simplified. We will see that latter, on Section 7.2.

3.7 Some Applications of Modal Logic

3.7.1 Epistemic Logic

In the next three chapters of this thesis, we will build formalisms to model, among other things, agents knowledge and beliefs. This will be done using ideas from epistemic logics. Therefore, it seems sensible to see a small introduction of these modal logics here. This introduction is very brief. The interested reader may consult, e.g., (van Ditmarsch et al. 2015; Fagin et al. 1995; Meyer and van der Hoek 1995) for more details.

The birth of *epistemic logics (EL)* is usually attributed to Hintikka (1962). These formalisms use possible worlds semantics to model knowledge and belief. Using the very same words of Halpern and Moses (1992): "the essential idea behind possible worlds semantics is that an agent's state of knowledge [or belief] corresponds to the extent to which he can determine what world he is in". Let a possible world be given, we associate, to each agent, a set of possible worlds that the gent considers it possible that she is in.

As we can see, this idea can be easily captured in modal logic, as follows. Let the vocabulary of epistemic logic be $\langle \mathbb{P}, \mathbb{A} \rangle$, where:

- $\mathbb{P} = \{p_0, p_1, \ldots\}$ is a non-empty set of propositional variables; and
- $\mathbb{A} = \{1, 2, \ldots, |\mathbb{A}|\}$ is a finite set of modalities representing the agents in the environment.

The environment itself is represented by an *epistemic model of knowledge*, which is a Kripke model $M = \langle W, R, I \rangle$, where W and I are as in Definition 3.4 and:

- $R : \mathbb{A} \to (W \times W)$ is a function that associates a reflexive and Euclidean indistinguishability relation to each agent $i \in \mathbb{A}$;

We have that, if the agent i is in the possible world w, all elements in $R_i(w)$ are indistinguishable from w for i.

The language of the *epistemic logic of knowledge* is $\mathcal{L}_{\mathrm{ML}}$, where the modal box operators are noted K_i, instead of $[m]$, just to recall that they mean knowledge.[7] Formulas

[6] In fact, in this specific case, an *inclusion test* must also be added to guarantee the termination of the method.

[7] This is just a notational variation turns out to be useful when we start mixing different kinds of modal operators. For instance, in the next section, when we see a formula of the form $[a]\mathrm{K}_i\varphi$, we immediately know that the second modal operator is the one of epistemic logic of knowledge. The diamond operator is not used often in this logic but, when it is, usually it is noted $\hat{\mathrm{K}}$.

of the form $\mathrm{K}_i\varphi$ are read 'agent i knows that φ'. We sometimes use $\mathcal{L}_{\mathrm{EL}}$ to refer to the language of epistemic logic.

The constraints imposed on each R_i imply that the axiom system of epistemic logic of knowledge is the one in Table 3.1 plus the following two (recall Table 3.2):

(T) $\mathrm{K}_i\varphi \to \varphi$ (knowledge)

(5) $\neg\mathrm{K}_i\varphi \to \mathrm{K}_i\neg\mathrm{K}_i\varphi$ (negative introspection)

In other words, epistemic logic of knowledge is $\mathrm{KT5}_n$ (which is also commonly known as $\mathrm{S5}_n$).

The semantics above is explicitly chosen to validate these axiom schemas. Axiom schema (T) stipulates that, if an agent knows that φ, then φ is true. This means that one cannot know something that is false, which is commonly considered to be the difference between knowledge and belief. Axiom schema (5) stipulates that, if an agent does not know that φ, then the agent knows that she does not know that φ.

It is also interesting to recall that the axiom system above validates the following (see Figure 3.2):

(D) $\neg\mathrm{K}_i\bot$ (consistency)

(4) $\mathrm{K}_i\varphi \to \mathrm{K}_i\mathrm{K}_i\varphi$ (positive introspection)

Axiom schema (D) stipulates that an agent does not know inconsistent facts. Axiom schema (4) stipulates that, if an agent knows that φ, then the agent knows that she knows that φ.

Example 3.2 (The Light Bulb)**.** As an example, let us imagine a closed room with a light bulb that is on. The state of the light is represented by the propositional variable *light*, which is true if and only if the light is on. Alice (agent 1) is inside the room and thus can see that the light is on. Betty (agent 2) is outside the room and thus cannot see the state of the light. Both agents know the location of each other. This environment can be represented by the epistemic model in Figure 3.5.

Alice can see the state of the light, which means that she can distinguish between w_0 and w_1. This is why worlds w_0 and w_1 are separated in the indistinguishability relation of Alice R_1. Betty, on the other hand, does not see the light bulb. This is why both worlds w_0 and w_1 are related in her indistinguishability relation R_2. Also note that it is indeed an epistemic model of knowledge, since both relations in R are equivalence relations.

In the actual world, the light is on. We have that $\langle M, w_0\rangle \vDash \mathit{light}$. In addition, Alice knows that the light is on. We have that $\langle M, w_0\rangle \vDash \mathrm{K}_1\mathit{light}$. Betty, does not know it, i.e., $\langle M, w_0\rangle \vDash \neg\mathrm{K}_2\mathit{light}$. In fact, Betty does not know the state of the light. She does not know whether it is on or off, i.e., $\langle M, w_0\rangle \vDash \neg\mathrm{K}_2\mathit{light} \wedge \neg\mathrm{K}_2\neg\mathit{light}$.

Both agents know the location of each other. Therefore, Alice must know that Betty does not know the state of the light. Indeed, we have $\langle M, w_0\rangle \vDash \mathrm{K}_1(\neg\mathrm{K}_2\mathit{light} \wedge \neg\mathrm{K}_2\neg\mathit{light})$. We also have that Betty knows the location of Alice and thus knows that Alice knows the state of the light. Indeed, we also have $\langle M, w_0\rangle \vDash \mathrm{K}_2(\mathrm{K}_1\mathit{light} \vee \mathrm{K}_1\neg\mathit{light})$. Finally, note that Betty does not know that Alice knows that the light is on, i.e., $\langle M, w_0\rangle \vDash \neg\mathrm{K}_2\mathrm{K}_1\mathit{light}$, nor that the light is off, i.e., $\langle M, w_0\rangle \vDash \neg\mathrm{K}_2\mathrm{K}_1\neg\mathit{light}$. ∎

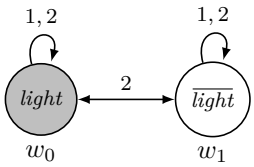


Figure 3.5 – An epistemic model of knowledge

We see in Example 3.2 that $\mathrm{KT5}_n$ correctly models agents knowledge. In addition, we saw that it can be used to model agents knowledge about other agents knowledge too. Let us see how to do the same with beliefs.

An *epistemic model of belief* is a Kripke model $M = \langle W, R, I\rangle$, where W and I are as in Definition 3.4 and:

- $R : \mathbb{A} \to (W \times W)$ is a function that associates a serial, transitive and Euclidean indistinguishability relation to each agent $i \in \mathbb{A}$;

As before, we have that, if the agent i is in the possible world w, then all elements in $R_i(w)$ are indistinguishable from w for i.

The language of the *epistemic logic of belief* is $\mathcal{L}_{\mathrm{ML}}$, where the modal box operators are noted B_i, instead of $[m]$, just to recall that they mean belief. Formulas of the form $\mathrm{B}_i\varphi$ are read 'agent i believes that φ'.

The constraints imposed on each R_i imply that axiom system of epistemic logic of belief is the one in Table 3.1 plus the following (again, recall Table 3.2):

(D) $\neg \mathrm{B}_i \bot$ (consistency)

(4) $\mathrm{B}_i\varphi \to \mathrm{B}_i\mathrm{B}_i\varphi$ (positive introspection)

(5) $\neg \mathrm{B}_i\varphi \to \mathrm{B}_i\neg\mathrm{B}_i\varphi$ (negative introspection)

In other words, the epistemic logic of belief is $\mathrm{KD45}_n$.

As for knowledge, the semantics of belief is chosen to validate these axiom schemas. Axiom schema (D) stipulates that an agent does not belief in inconsistent facts. Axiom schema (4) stipulates that, if an agent believes φ, then she believes that she believes φ. And finally, axiom schema (5) stipulates that, if an agent does not believe φ, then she believes that she does not believe φ.

Example 3.3 (The Light Bulb Revisited)**.** Let us imagine again the environment of Example 3.2. But, now Alice leaves the room without telling Betty. The state of the light inside the room may have changed, so Alice does not know if it is on or off anymore. Betty still thinks that Alice is inside the room. This can be modeled with the epistemic model in Figure 3.6.

Alice does not know the state of the light. This is why she cannot distinguish between w_0 and w_1. The situation for Betty is different. In the actual world w_0, she thinks that Alice is in the room. Therefore, in w_0 Betty thinks that she is either in w_2 or in w_3, where Alice can distinguish between the states where the light is on and off. Note that the model satisfies all the required constraints for each indistinguishability relation R_i.

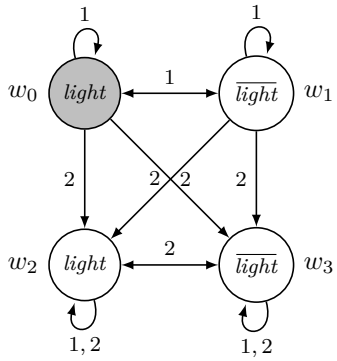


Figure 3.6 – An epistemic model of belief

In the actual world the light is still on, thus we have that $\langle M, w_0\rangle \vDash \mathit{light}$. Alice cannot be sure of the state of the light anymore. Thus Alice does not believe it, i.e., $\langle M, w_0\rangle \vDash \neg \mathrm{B}_1 \mathit{light}$. In fact, Alice now has two possibilities, either the light is on or off, i.e., $\langle M, w_0\rangle \vDash \mathrm{B}_1(\mathit{light} \vee \neg \mathit{light})$. Betty still believes that Alice is sure about the state of the light. That is, $\langle M, w_0\rangle \vDash \mathrm{B}_2(\mathrm{B}_1 \mathit{light} \vee \mathrm{B}_1 \neg \mathit{light})$. Alice did not tell Betty that she left the room. Thus Alice believes that Betty still believes that she is sure of the state of the light, i.e., $\langle M, w_0\rangle \vDash \mathrm{B}_1\mathrm{B}_2(\mathrm{B}_1 \mathit{light} \vee \mathrm{B}_1 \neg \mathit{light})$. ∎

A complete exposition of epistemic knowledge usually also presents operators modeling group knowledge, distributed knowledge and common knowledge. To keep this introduction brief, we do not present them here (but we do introduce a group knowledge operator in Chapter 4). Many extensions of epistemic logic exist. Operators modelling agents abilities, actions and time are among them. For a survey of such extensions, the reader may consult (van Ditmarsch et al. 2015).

3.7.2 Dynamic Epistemic Logic

One of the extensions of epistemic logic is *dynamic epistemic logic (DEL)*, originally proposed by Baltag and Moss (2004) and Baltag et al. (1998). We will meet this logic on chapters 5 and 6. This is why we make a very brief introduction here. For more information, please consult (van Ditmarsch, van der Hoek, et al. 2007).

Dynamic epistemic logic extends EL with operators aiming at modelling the dynamics of knowledge. It adds to EL operators of the form $[e]\varphi$, where e is a possible event. A formula of the form $[e]\varphi$ is read 'after every possible occurrence of e it is the case that φ'.

DEL defines a structure called *event model* containing all possible events which may modify the knowledge of the agents.[8]

[8] In fact, the original DEL uses a finite set of event models. But all the event models can be grouped together in a single event model, as we do here. Both definitions are equivalent. The addition of post-conditions is originally from (van Ditmarsch et al. 2005).

Definition 3.17 (Event Model)**.** Let a vocabulary $\langle \mathbb{P}, \mathbb{A} \rangle$ be given. An *event model* is a structure $N = \langle E, P, \text{pre}, \text{post} \rangle$, where:

- E is a set of possible events;
- $P : \mathbb{A} \to (E \times E)$ is a function that associates an indistinguishability relation to each agent $i \in \mathbb{A}$.
- $\text{pre} : E \to \mathcal{L}_{\text{EL}}$ is a function that associates a pre-condition $\varphi \in \mathcal{L}_{\text{EL}}$ to each event $e \in E$.
- $\text{post} : E \to (\mathbb{P} \to \mathcal{L}_{\text{EL}})$ is a function that associates a partial post-condition function $\text{post}(e)$ to each event $e \in E$.

Therefore, an event model is a special kind of epistemic model. it has possible events instead of possible worlds, indistinguishability relations between events, instead of between worlds and it has pre- and post-conditions for events instead of interpretations.

An example of event model is depicted in Figure 3.7a. In that picture, agent 1 can distinguish between e_0 and e_1, while agent 2 cannot. This means that, if e_0 is the actual event occurring, then agent 1 is able to know that it is occurring, whereas agent 2 cannot make that distinction. Each event in the picture contains a pair $(\text{pre}, \text{post})$ representing its pre- and post-condition, respectively. For instance, the pre-condition of e_0 is *light* while its post-condition is $\emptyset$, meaning that this event occurs only if *light* is true and that it does produce any factual change.

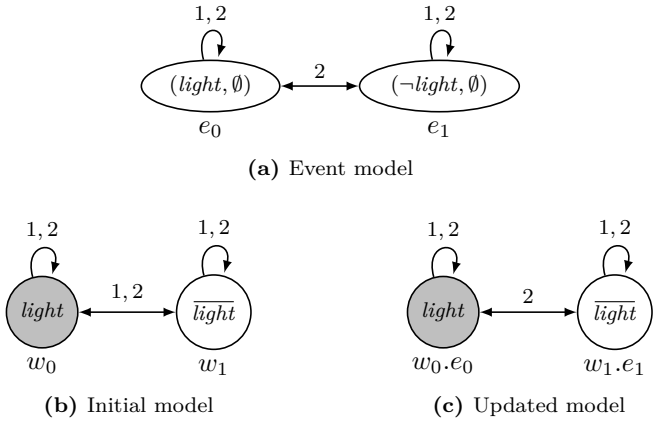
The occurrence of an event e in a pointed epistemic model $\langle M, w \rangle$, changes the pointed epistemic model. The result is a new pointed epistemic model, which is obtained from the *product* between the epistemic model and the event model, defined as follows.

Definition 3.18 (Model Product)**.** The *product* between an epistemic model $M = \langle W, R, I \rangle$ and an event model $N = \langle E, P, \text{pre}, \text{post} \rangle$ is a new epistemic model $M.N = \langle W^N, R^N, I^N \rangle$, where:

- $W^N = \{w.e \mid M, w \vDash \text{pre}(e)\}$
- $R_i^N = \{(w.e, w'.e') \mid (w, w') \in R_i \text{ and } (e, e') \in P_i\}$
- $I_p^N = \begin{cases} \{w.e \mid \langle M, w \rangle \vDash \text{post}(e)(p)\}, & \text{if } \text{post}(e)(p) \text{ is defined} \\ I_p, & \text{otherwise} \end{cases}$

Intuitively, the product between M and N is a Kripke model $M.N$ where each of its possible worlds $w.e$ is a combination of a possible world w of M and a possible event e of N, provided that the pre-condition of e is true at w. The indistinguishability relation of the resulting model is calculated using the relations in M and N. The interpretations in the resulting model are calculated using the post-conditions post in N.

Definition 3.19 (Update)**.** Let an event model $N = \langle E, P, \text{pre}, \text{post} \rangle$ be given. The update of a pointed epistemic model $\langle M, w \rangle$ by an event $e \in E$ is is the new pointed epistemic model $\langle M.N, w.e \rangle$.



(a) Event model

(b) Initial model

(c) Updated model

Figure 3.7 – Model product

Example 3.4 (Prelude to The Light Bulb)**.** Let us get back to Example 3.2 but, this time, before Alice enters the room. We have that both Alice and Betty are ignorant about the state of the light in the room, and they know each other locations. This can be modeled by the epistemic model in Figure 3.7b.

Now, suppose that Alice enters the room. Alice observes the state of the light, Betty knows that Alice observes the state of the light, but she does not know if Alice observes that the light is on or off. This is modeled by the event model in Figure 3.7a.

It is important to grasp the idea that the pre-condition of an event can also be seen as the observation made by the agents when this is the actual event occurring. Indeed, agents cannot observe that φ is true, if φ is actually false. This is why it is a pre-condition.

In the model in Figure 3.7a, we have that the pre-condition of e_0 is *light*, meaning that, when this event occurs, the agents observe that the light is on on the room. The pre-condition of e_1 is $\neg$*light*, meaning that, when this event occurs, the agents observe that the light is off in the room. Therefore, we have that Betty does not observe the state of the light, but she does observe that Alice observes it.

The resulting model is depicted in Figure 3.7c. Note that it is bisimilar to the model in Figure 3.5. ∎

Event models and updates are a powerful tool. The number of different events that can be modeled is huge (it is infinite, actually). For instance, we will see in Chapter 6 that we use this to model belief expansion and belief revision.

There are some limitations nonetheless. Note that the indistinguishability relations P_i in the event model of Example 3.4 are all equivalence relations. This is necessary. If relations P_i are not equivalence relations, the result of the product may not be a $\mathrm{KT5}_n$ model. There are similar restrictions must be made when working with $\mathrm{KD45}_n$ models.

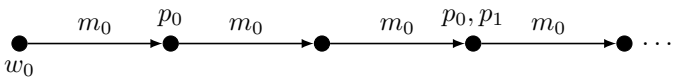


Figure 3.8 – A temporal model

3.7.3 More applications

In fact, modal logics have a huge number of applications. We could keep listing them for a long time. In this section, we just briefly mention two more.

The model in Figure 3.8 represents a temporal model, as in the modal logic called linear temporal logic (LTL). In this logic, possible worlds are seen as instants in time. Assume that we define $R(t)$ as the transitive closure of $R(m_0)$. We have that the modal operator $\langle t \rangle$ means 'eventually' and the operator $[t]$ means 'always'. For example, $\langle M, w_0 \rangle \vDash \langle t \rangle (p_0 \wedge p_1)$, meaning that $(p_0 \wedge p_1)$ is 'eventually' true in the future, and $\langle M, w_0 \rangle \nvDash [t] p_0$ means that p_0 is not 'always' true in the future.

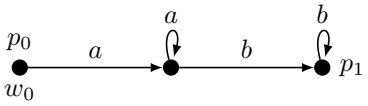


Figure 3.9 – A dynamic model

Another application that interests us in this thesis is the one exemplified in Figure 3.9. This is a model of propositional dynamic logic (PDL). In this formalism, possible worlds are states and the relations are state transitions. For example, $\langle M, w_0 \rangle \vDash p_0 \wedge [a][b] p_1$ means that, the initial state w_0 satisfies p_0 and the execution of the sequence of actions $a; b$ invariably leads to a state satisfying p_1.

3.8 Conclusion

In this chapter, we saw a brief introduction to modal logic. We saw the basics, its syntax, semantics, axiom systems and a tableau method. Some applications have also been presented.

From the next chapter on, we will put all the introductory material seen up till now at work. The next two chapters present modal logics of action and knowledge. Then, we will see modal logics for belief and belief revision. After that, we will see new methods for modal logic satisfiability checking.

Chapter 4

A Modal Logic of Responsibility

This chapter presents a formalism aiming at modeling individual and collective responsibility, and also the *problem of many hands*. This is a problem that arises whenever an organisation is responsible for some undesirable outcome, but none of its members can be held responsible for that outcome. The formalism proposed here is a logic that brings together notions of enacted actions, agents abilities, group knowledge and organisational structures. A sound and complete axiom system for the formalism is provided, as well as formal definitions for individual and collective responsibility, and for the problem of the many hands.

This is a synthesis of a series of publications on the subject (de Lima et al. 2010a; de Lima and Royakkers 2015; de Lima et al. 2010b). The first section below presents and discusses the problem of many hands, which is the motivation of the work. The formal logic, its syntax, semantics and axiom system are presented on Section 4.2. After that, Section 4.3 uses that formalism to present formal definitions for individual and collective responsibility. These definitions are in turn used to formalise the problem of many hands in Section 4.4. Section 4.5 discusses related work and Section 4.6 concludes the chapter.

4.1 Motivation

The term *problem of many hands* (hereafter PMH) has been coined by Thompson (1980) and is meant to designate a situation where:

> "a group of individuals can reasonably be held responsible for an undesirable outcome, while no member of the group can reasonably be held responsible for the outcome."

It is not evident whether such a situation may actually occur. Indeed, one may tend to ascribe responsibility to a group only if at least some member of the group is

#votes	a_1	a_2	a_3
3	no	yes	yes
3	yes	no	yes
3	yes	yes	no
Decision:	yes	yes	yes

Table 4.1 – Agents votes on each action of Example 4.1

responsible. However, we argue that the PMH can occur in practice by giving here three examples.

Example 4.1. The first example is inspired by the *doctrinal paradox* (Kornhauser and Sager 1986; Petit 2001). We suppose that a group of 9 agents have to decide whether the actions a_1, a_2 and a_3 will be performed. They decide it using a simple majority election for each of the actions. Every agent knows that an undesirable outcome φ is brought about if all the three actions are performed. The way the agents vote on this issue is depicted on Table 4.1. Note that no agent agrees that all the three actions should be performed. We therefore assume that no agent intends to obtain φ. However, the final result is such that all the three actions are performed which means that the undesirable outcome φ is obtained. Also note that, for each agent taken individually, if this agent changes its vote, it does not change the final result.

We conclude that, when assuming that the vote is secret and that there is no previous arrangement between the agents, no agent is, individually, responsible for φ, whereas the group is responsible for that. Hence, a PMH. ∎

Example 4.2. In the second example, we assume an environment with 4 agents where each one has the ability to execute a different action: agent 1 is the only one that can execute action a_1; agent 2 is the only one that can execute action a_2; and so on. The execution of action a_i by agent i brings a desirable outcome for this agent. But every agent knows that the execution of three different actions a_i brings about an undesirable outcome φ.

Here, the agents do not vote on the actions, they simply decide, individually, whether to perform the action or not. If we assume that the agents are rational, the decision taken by each of them depends on the reward of the outcomes involved. For each agent, if the reward of the outcome of its own action outweighs the utility of φ, then the agent decides to perform its own action. If so is the case for all agents, we have that the undesirable outcome φ is brought about. Table 4.2 depicts agents decisions in this latter case. Again, note that if one agent, taken individually, changes its decision, it is not enough to change the result.

Since no agent, individually, bought about φ, we cannot ascribe to one agent the responsibility for φ. However, the group acted in a way that brought about φ. Hence, again, a PMH. ∎

Example 4.3. The third example is inspired by the *prisoners dilemma* (Kuhn 2019). In this version, we assume an environment with two agents. Each one has the ability to execute the two actions a_1 and a_2. Both agents know that if they do not execute the

agent	a_1	a_2	a_3	a_4
1	yes	–	–	–
2	–	yes	–	–
3	–	–	yes	–
4	–	–	–	yes

Table 4.2 – Agents decisions on Example 4.2

	a_1	a_2
a_1	OK	undesirable
a_2	undesirable	OK

Table 4.3 – Possible outcomes on Example 4.3

same action, an undesirable outcome φ is brought about. This situation is depicted in Table 4.3. The agents cannot communicate.

Each agent has no way to know what the other agent decides to do. Then, if the undesirable outcome is obtained, each agent, individually, cannot be hold responsible for it. But the group act in a way such that φ is obtained. Hence, yet again, a PMH. ■

In these three examples, one may argue that each agent *shares* the responsibility for φ. This would mean that the agents are not completely responsible for the undesirable outcome, but they are responsible for it to some degree. For instance, in the first example, the agent not willing to share responsibility for φ should simply vote 'no' in all columns.

The way we see things, there is a problem with this argument. A rational agent has no reason to think that the other agents would vote in a different way. Remember that the assumption made here is that the agents cannot talk to each other. The agent cannot foresee that the other agents will vote differently and thus cannot foresee that the undesirable outcome will be brought about. We believe that an agent cannot be held responsible for what cannot be foreseen. In the second and third examples, something similar happens. Each agent does not know the reward of the actions for the other agents. So, again, the agent cannot foresee the outcome and cannot be held responsible.

According to Bovens (1998), the PMH can happen and when it does, it raises a problem when people are harmed, because:

> it "frustrates the need for compensation and retribution on the part of the victims",

and also:

> "the fact that no one can be meaningfully called to account after the event also means... that no one need feel responsible beforehand".

But we overlooked an important thing in the above discussion. We did not precise what we mean by the term 'responsibility'. For example, when we say that someone is

responsible for an outcome φ, it may mean that one is 'accountable' for φ or that one is 'blamed' for φ, or even, that one has the 'obligation to see to it that' φ. van de Poel et al. (2015) define several different meanings for the term 'responsibility' using other, more basic, concepts, such as moral agency, causality, wrong-doing, freedom and knowledge:

- Descriptive meanings:
  - Responsibility-as-cause:
    "The earth quake is responsible for the death of 100 people."
  - Responsibility-as-task:
    "The train driver is responsible for driving the train."
  - Responsibility-as-authority:
    "The project manager is responsible for the project."
  - Responsibility-as-capacity:
    (The ability to act in a responsible way.)
- Normative forward-looking meanings:
  - Responsibility-as-virtue:
    "She is a responsible person."
  - Responsibility-as-obligation:
    "The bus driver is responsible for the safety of the passengers."
- Normative backward-looking meanings:
  - Responsibility-as-accountability:
    (The obligation to account for one's actions and theirs outcomes.)
  - Responsibility-as-blameworthiness:
    "The driver is responsible for the car accident."
  - Responsibility-as-liability:
    "He is liable to pay damages."

In this work, we are interested on some normative meanings of responsibility, namelly, obligation, accountability and blameworthiness. For instance, an individual i is accountable for φ if i is capable of acting as a moral agent, behaves in a way that is not morally acceptable (i.e., does something wrong) and this behavior causes φ. In addition, i is blamed for φ if i is accountable for φ, knows (or could know) that φ would be the case, and acts freely, i.e., i can chose to behave differently, and this different behavior avoids φ. Finally, i has the obligation to see to it that φ if i has the obligation that φ and i must in fact make φ occur, i.e., i has the obligation to cause φ.

These different meanings of responsibility are also classified in two categories. Accountability and blameworthiness are 'backward-looking responsibilities', i.e., responsibilities regarding the past, while 'obligation to see to it that' is a 'forward-looking responsibility', i.e., a responsibility regarding the future. Van de Poel et al. also argues that these three notions are related. Roughly, if an individual i has the obligation to see to it that φ but does not obtain φ, then i is accountable for $\neg\varphi$ and, if i does not provide a satisfactory account for $\neg\varphi$, then i is blamed.

4.2 The Formal Framework

In this section we present a logic that will be the basis of our formalization of individual and collective responsibility. This logic is a variation of the coalition epistemic dynamic logic (CEDL) presented in (de Lima et al. 2010b), which, by its turn, is an extension of the well-known propositional dynamic logic (PDL) (Harel 1984; Harel et al. 2000). PDL is a classical propositional logic augmented with modal operators '$[a]$'. A formula of the form $[a]\varphi$ means 'after every possible occurrence of event a, the consequence φ is true'. Thus, it permits expression of what consequences are caused by the occurrence of some given event a, where such events can be actions executed by one agent, actions executed by several agents, exogenous events or even programs. But since PDL does not have agents in its language, it does not enable expression of what consequences are caused by which agents of the scenario. To be able to express agent causality, CEDL extends it by actions that are 'enacted' by agents, in a similar way as done, e.g., by Royakkers (1998) and Wieringa and Meyer (1993) and also more recently by Herzig and Lorini (2010a,b). In CEDL, one can write formulas of the form $[(i, a)]\varphi$, meaning 'after every possible occurrence of event (i, a), the consequence φ is true', where the event (i, a) is the action a executed by agent i. Moreover, to be able to express agent knowledge, CEDL has modal operators K_i, in a similar way as done, e.g., by Grossi et al. (2007) and Herzig et al. (2000). In CEDL, one can also write formulas of the form $\mathrm{K}_i\varphi$, meaning 'agent i knows that φ'.

The resultant formalism is a logic presenting some of the properties of PDL and epistemic logic (Fagin et al. 1995). In addition, because the actions of CEDL are enacted by agents, it is possible to define operators expressing agent abilities and agent obligations. It turns out that these operators are similar to the ones in coalition logic (Pauly 2001, 2002), alternating-time temporal logic (Alur et al. 2002) and alternating-time temporal epistemic logic (van der Hoek and Wooldridge 2003). The obligation operators are similar to the ones in dynamic deontic logic (d'Altan et al. 1996; Meyer 1988; Meyer and Wieringa 1993). Therefore, CEDL constitutes a vary expressive framework where one can express actions, knowledge, abilities and obligations, all at once.

4.2.1 Models

A model is defined for a given vocabulary, which, in this case, consists of a triple of disjoint sets $\langle \mathbb{P}, \mathbb{A}, \mathbb{T} \rangle$, where:

- $\mathbb{P}$ is a countable (possibly infinite) set of propositional variables denoting propositional facts;
- $\mathbb{A}$ is a finite set of labels denoting agents; and
- $\mathbb{T}$ is a finite set of labels denoting the actions available for the agents.

We use Δ to denote the set of all joint actions available for the agents in $\mathbb{A}$, which is defined as the set of all total functions δ with signature $\mathbb{A} \to \mathbb{T}$. In other words, $\Delta = \{\delta_1, \delta_2, \ldots\}$, where each δ_n is a set of pairs of the form $\{(i_1, a_1), (i_2, a_2), \ldots, (i_{|\mathbb{A}|}, a_{|\mathbb{A}|})\}$ (one pair for each agent in $\mathbb{A}$) and where $i_m \in \mathbb{A}$ and $a_m \in \mathbb{T}$.

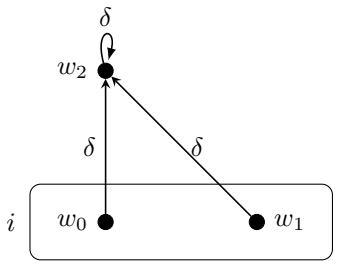


Figure 4.1 – Graphic representation of CEDL models: dots represent possible worlds, arrows represent transitions and rectangles represent accessibility relations equivalence classes. Possible worlds outside rectangles are alone in their class.

CEDL models are a specific kind of Kripke models, where nodes represent possible worlds and relations between worlds represent either accessibility relations or transitions. A graphical representation of such models is displayed in Figure 4.1.

The formal definition is as follows. Let the vocabulary $\langle \mathbb{P}, \mathbb{A}, \mathbb{T} \rangle$ be given, a CEDL model is a quadruple $\langle W, R, T, I \rangle$, where:

- W is a non-empty set of possible worlds;
- R is a function with signature $\mathbb{A} \to (W \times W)$, defining, for each agent $i \in \mathbb{A}$, an equivalence relation between possible worlds, which represents the knowledge of agent i;
- T is a function with signature $\Delta \to (W \times W)$, defining, for each joint action $\delta \in \Delta$, a relation between possible worlds, which represents the transition associated to the joint action δ; and
- I is a function with signature $\mathbb{P} \to 2^W$, defining, for each $p \in \mathbb{P}$, the interpretation of the propositional variable p in the model.

To simplify notation, we sometimes write R_i instead of $R(i)$, and also use $R_i(w)$ to denote the set of worlds that i considers possible at w. Analogously, we sometimes write T_δ instead of $T(\delta)$, and also use $T_\delta(w)$ to denote the set of possible outcomes of the occurrence of δ at w.

Some assumptions are implicit in the definition of CEDL models. For instance, every relation R_i is an equivalence relation, i.e., they are all reflexive, transitive and euclidean. These are standard assumptions when modeling the knowledge of agents. As we have seen in Chapter 2, these assumptions correspond to axioms T, 4 and 5. Therefore, we have the following properties in CEDL:

- Truth: if agent i knows that φ is true, then φ is true;
- Positive introspection: if agent i knows that φ is true, then i knows that i knows that φ is true; and

- Negative introspection: if agent i does not know that φ is true, then i knows that i does not know that φ is true.

Even with a small vocabulary, structures respecting the definition above can model many different scenarios. Some of them are interesting, but some may be considered strange, if not useless. For instance, the definition above does not forbid structures where, for some possible world $w \in W$ and for all $\delta \in \Delta$, we have $T_\delta(w) = \emptyset$. This is considered a strange structure because we cannot conceive a scenario that could be modeled in such a way. We consider here that an action is every behavior that takes time. This means that even "doing nothing" is an action. In fact, this is a special kind of action that does not change the state of affairs, but that has an outcome and therefore has a non-empty transition leading to a possible world: one which satisfies the same propositional variables as before the execution of the action. We then conclude that there is always some action available for the agents in every possible world. Therefore, we impose it to CEDL models. This is done by imposing the following constraint:

(C1) for all $w \in W$ there is $\delta \in \Delta$ such that $T_\delta(w) \neq \emptyset$

Constraint C1 is called 'activity'. It stipulates that, at all possible worlds of the model, there is at least one non-empty transition which is labeled by some joint action in Δ. In other words, at any moment, there is at least one executable action for each agent, which prevents those strange structures mentioned earlier from being CEDL models.

There is yet another constraint imposed to CEDL models:

(C2) for all $w \in W$, all $i \in \mathbb{A}$ and all $\delta \in \Delta$, we have $(T_\delta \circ R_i)(w) \subseteq (R_i \circ T_\delta)(w)$

This constraint corresponds to 'no-forgetting' in (Herzig et al. 2000) and 'perfect recall' in (Fagin et al. 1995). It defines an interaction between accessibility relations and transitions. With this constraint, the knowledge of an agent either increases or stays the same, after the execution of any action. This means that agents never loose information, i.e., once an agent knows something, this agent will never forget it. This is obviously very restrictive but, at the same time, useful. For instance, it avoids to consider models where agents may keep losing information for whatever reason. Such possibility would make much more difficult (if not impossible) to derive some interesting properties of our formalism. We also note that, because each $R(i)$ is an equivalence relation, Constraint C2 implies that action occurrences are perceived by all agents. The latter implies that the agents perceive the passage of time.

For example, the structure in Figure 4.1 respects both C1 and C2. The first is respected because, for all possible worlds w, there is at least one transition from w to some possible world. The second is respected because the equivalence classes of the accessibility relation R_i do not increase their size when we follow a transition from one possible world to another.

4.2.2 Syntax and Semantics of CEDL

Let the vocabulary $\langle \mathbb{P}, \mathbb{A}, \mathbb{T} \rangle$ be given, the language of CEDL, $\mathcal{L}_{\text{CEDL}}$, is defined by the BNF:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{K}_i\varphi \mid [\delta|_G]\varphi$$

where $p \in \mathbb{P}$, $i \in \mathbb{A}$, $G \subseteq \mathbb{A}$ and $\delta \in \Delta$. The construction $\delta|_G$ is δ, but with its domain restricted to G, i.e., let δ be the joint action $\{(i_1, a_1), \ldots, (i_{|\mathbb{A}|}, a_{|\mathbb{A}|})\}$, then $\delta|_G$ is the partial joint action formed by the set of pairs $\{(i_n, a_n) \mid (i_n, a_n) \in \delta \text{ and } i_n \in G\}$.

In what follows, the common abbreviations for the operators $\wedge$, $\rightarrow$ and $\leftrightarrow$ are also used, the symbol $\bot$ (contradiction) abbreviates $\neg\top$, and, to simplify notation, we sometimes write $[i_1{:}a_1, \ldots, i_n{:}a_n]\varphi$ instead of $[\{(i_1, a_1), \ldots, (i_n, a_n)\}]\varphi$.

The fragment without operator $[\,]$ is, in fact, multi-modal KT5, where each modality $m \in \mathbb{M}$ is identified with an agent $i \in \mathbb{A}$. The modal operator K_i is used instead of $[m]$ to stress its "knowledge" meaning.

The intended meaning of formulas of the form $\mathrm{K}_i\varphi$ is, as usual: 'agent i knows that φ'. The intended meaning of a partial joint action of the form $\{(i_1, a_1), \ldots, (i_n, a_n)\}$ is: 'the agents in $\{i_1, \ldots, i_n\}$ execute their corresponding actions in $\{a_1, \ldots, a_n\}$ simultaneously (and we do not consider what the other agents do at the same time)'. And the intended meaning of formulas of the form $[\delta|_G]\varphi$ is: 'after every possible occurrence of δ_G, φ is true'.

To match their intended meanings, formulas from language $\mathcal{L}_{\mathrm{CEDL}}$ are interpreted using 'pointed CEDL models'. The latter are pairs of the form (M, w), where $M = \langle W, R, T, I\rangle$ is a CEDL model and $w \in W$. Then, the semantic interpretation of Boolean operators is the usual one. For formulas of the form $\mathrm{K}_i\varphi$ we use the accessibility relation labeled with i: $\mathrm{K}_i\varphi$ is true at the world w if and only if φ is true at all possible worlds w' accessible from w via the accessibility relation labeled with i, i.e., it is true if and only if φ is true at all worlds that the agent i considers possible at w. The interpretation of operators $[\delta|_G]$ is more complex. A formula of the form $[\delta|_G]\varphi$ is true at the world w if and only if φ is true at all possible worlds w' that are attained from w via some transition in T which is labeled with $\delta|_G \cup \delta'|_{\mathbb{A}\setminus G}$. In other words, to verify whether $[\delta|_G]\varphi$ is true at some possible world w, we must verify whether φ is true at all worlds w' belonging to $T_{\delta'}(w)$ for all $\delta' \in \Delta$, provided that the restriction of δ' to G is equal to the restriction of δ to G, i.e., provided that $\delta'|_G = \delta|_G$. That is, we must verify whether is true at all transitions from no matter what the agents outside the group do.

Formally, the satisfaction relation $\models$, between pointed CEDL models and formulas from $\mathcal{L}_{\mathrm{CEDL}}$, is recursively defined as follows:

$$
\begin{array}{lll}
\langle M, w\rangle \models \top & & \\
\langle M, w\rangle \models p & \text{iff} & w \in I(p) \\
\langle M, w\rangle \models \neg\varphi & \text{iff} & \langle M, w\rangle \not\models \varphi \\
\langle M, w\rangle \models \varphi \wedge \psi & \text{iff} & \langle M, w\rangle \models \varphi \text{ and } \langle M, w\rangle \models \psi \\
\langle M, w\rangle \models \mathrm{K}_i\varphi & \text{iff} & \text{for all } w' \in R_i(w) \text{ we have } \langle M, w\rangle' \models \varphi \\
\langle M, w\rangle \models [\delta|_G]\varphi & \text{iff} & \text{for all } \delta' \in \Delta \text{ and all } w' \in T_{\delta|_G \cup \delta'|_{\mathbb{A}\setminus G}}(w) \text{ we have } \langle M, w\rangle' \models \varphi
\end{array}
$$

We note that $\delta|_G \cup \delta'|_{\mathbb{A}\setminus G}$ is a set of pairs belonging to Δ. Thus, $T_{\delta|_G \cup \delta'|_{\mathbb{A}\setminus G}}$ is defined for all $w \in W$.

As usual, a formula φ is valid (notation: $\models \varphi$) if and only if every pointed CEDL model $\langle M, w\rangle$ satisfies φ.

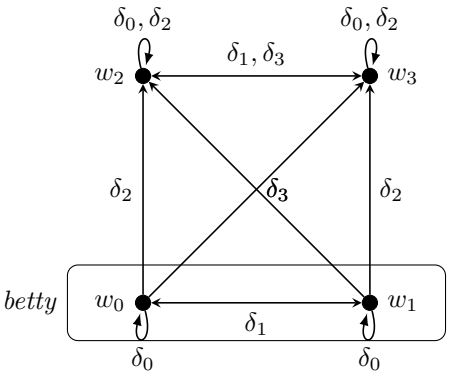


Figure 4.2 – A CEDL model for the Light Bulb and Light Switch scenario. (The rectangles represent Betty's knowledge. To simplify the picture, Alice's knowledge is not represented. She has complete knowledge of the scenario, i.e., $R_{alice}(w) = \{w\}$ for all $w \in W$.)

Example 4.4 (Light Bulb and Light Switch)**.** To better explain the definitions given so far, we now use a scenario that we call 'light bulb and light switch'. This scenario is inhabited by two agents: Alice (agent *alice*) and Betty (agent *betty*). They live in a strange house: its interior is illuminated by a light bulb, but the corresponding switch is located outside the house. In this scenario, Alice is inside the house and Betty is outside it, close to the switch. Thus, Alice can see whether the light bulb is on (noted *light*) or off (noted $\neg$*light*) and tell (or rater shout) it to Betty (action *tell*), but she cannot toggle the switch. Betty, on the other hand, can toggle the switch (action *tog*) but she cannot see whether the light is on or off. If she toggles the switch with the light on, it will turn off, and if she toggles it with the light off, it will turn on. Let the vocabulary be $\langle \mathbb{P}, \mathbb{A}, \mathbb{T} \rangle$, where $\mathbb{P} = \{light\}$, $\mathbb{A} = \{alice, betty\}$ and $\mathbb{T} = \{none, tell, tog\}$. A CEDL model for this scenario can be given by the structure in Figure 4.2.

We assume that the light bulb is off, i.e., the actual world is w_0. We then use the definition of $\models$ to verify truth of some formulas in the pointed CEDL model $\langle M, w_0 \rangle$:

- $\langle M, w_0 \rangle \models \neg light$.
  In words, the light is off. This is true because $w_0 \notin I(light)$, i.e., *light* is not in the label of w_0.

- $\langle M, w_0 \rangle \models \mathrm{K}_{alice} \neg light$.
  In words, Alice knows that the light is off. This is true because $M, w \models \neg light$ for all $w \in R_{alice}(w_0)$. (Recall that $R_{alice}(w_0) = \{w_0\}$.)

- $\langle M, w_0 \rangle \models \neg \mathrm{K}_{betty} light \wedge \neg \mathrm{K}_{betty} \neg light$.
  In words, Betty does not know that the light is on and she also does not know that the light is off. This is true because, first, there is a possible world that Betty considers possible (namely w_0) where the light is off and there is a possible world that Betty considers possible (namely w_1) where the light is on.

- $\langle M, w_0\rangle \models [\mathit{alice}{:}\mathit{none}, \mathit{betty}{:}\mathit{tog}]\mathit{light}$.
  In words, after the parallel execution of Alice doing noting and Betty toggling the switch, the light is on. This is true because, first, $\delta_1 = \mathit{alice}{:}\mathit{none}, \mathit{betty}{:}\mathit{tog}$. Second, $\mathbb{A} = \{\mathit{alice}, \mathit{betty}\}$, thus, for all $\delta' \in \Delta$, we have $\delta'|_{\mathbb{A}\setminus\{\mathit{alice},\mathit{betty}\}} = \delta'|_{\emptyset} = \emptyset$; Then, $\delta_1|_{\{\mathit{alice},\mathit{betty}\}} \cup \delta' = \delta_1$, which means that $T_{\delta_1|_{\{\mathit{alice},\mathit{betty}\}}\cup\delta'|_{\mathbb{A}\setminus\{\mathit{alice},\mathit{betty}\}}}(w_0) = T_{\delta_1}(w_0) = \{w_1\}$, for all $\delta' \in \Delta$, and finally, $M, w_1 \models \mathit{light}$.

- $\langle M, w_0\rangle \models [\mathit{alice}{:}\mathit{tell}, \mathit{betty}{:}\mathit{tog}]\mathit{light}$.
  In words, after the parallel execution of Alice telling whether the light is on or off and Betty toggling the switch, the light is on. Similarly as before, we have that $\delta_3 = \mathit{alice}{:}\mathit{tell}, \mathit{betty}{:}\mathit{tog}$ and $T_{\delta_3|_{\{\mathit{alice},\mathit{betty}\}}\cup\delta'|_{\mathbb{A}\setminus\{\mathit{alice},\mathit{betty}\}}}(w_0) = T_{\delta_3}(w_0) = \{w_3\}$, for all $\delta' \in \Delta$, and also $M, w_3 \models \mathit{light}$.

- $\langle M, w_0\rangle \models [\mathit{betty}{:}\mathit{tog}]\mathit{light}$.
  In words, after Betty toggling the switch, the light is on. (Note that here we consider only the action executed by Betty.) This is true if and only if, for every action that Alice can execute at the same time as Betty, after its parallel execution with Betty's action *tog*, the light is on. We have just verified, in the two preceding items, that if Alice does nothing this is indeed the case, as well as if Alice tells Betty whether the light bulb is on or off. But these are the only two options Alice has. Therefore, the sentence 'after Betty toggling the switch, the light is on' is true. More formally, note that $\delta_1|_{\{\mathit{betty}\}} = \mathit{betty}{:}\mathit{tog}$ and also $\delta_3|_{\{\mathit{betty}\}} = \mathit{betty}{:}\mathit{tog}$. Then, we just have to verify that $\langle M, w_0\rangle \models [\delta_1]\mathit{light}$ and also $\langle M, w_0\rangle \models [\delta_3]\mathit{light}$. But, this is what has been done in the two preceding items.

- $\langle M, w_0\rangle \models \neg[\mathit{betty}{:}\mathit{tog}]\bot$.
  In words, it is not the case that, after Betty toggling the switch, we have a contradiction. Since the formula $\bot$ is false in any possible world (by definition), then the latter sentence is equivalent to: 'it is not the case that the execution of Betty toggling the switch is not possible'. To show that it is true, we have to verify that $\langle M, w_0\rangle \models \neg[\delta_1]\bot$ or $\langle M, w_0\rangle \models \neg[\delta_3]\bot$. Both are the case, because $M, w \models \neg\bot$ for all $w \in W$ (by definition).

- $\langle M, w_0\rangle \models \neg[\mathit{alice}{:}\mathit{tell}]\mathrm{K}_{\mathit{betty}}\neg\mathit{light}$.
  In words, it is not the case that after Alice telling whether the light bulb is on or off to Betty, she knows that the light is off. This is the case because, if Betty toggles the switch at the same time, the light will be on. If it is on, then it is not possible for Betty to know that this is off. (Note that it uses the assumption that agents cannot know false statements.) Indeed, we have that $\langle M, w_0\rangle \not\models [\mathit{alice}{:}\mathit{tell}, \mathit{betty}{:}\mathit{tog}]\mathrm{K}_{\mathit{betty}}\neg\mathit{light}$. To show it, we first note that $\delta_3 = \mathit{alice}{:}\mathit{tell}, \mathit{betty}{:}\mathit{tog}$. Moreover, $T_{\delta_3}(w_0) = \{w_3\}$. And also, $M, w_3 \models \neg\mathrm{K}_{\mathit{betty}}\neg\mathit{light}$. The latter is the case because $R_{\mathit{betty}}(w_3) = \{w_3\}$ and $w_3 \in I(\mathit{light})$.

Many other interesting (and more complex) formulas can be verified in this model. For instance, we leave it to the reader to verify that the following is true: $\langle M, w_0\rangle \models [\mathit{alice}{:}\mathit{tell}, \mathit{betty}{:}\mathit{none}]\mathrm{K}_{\mathit{betty}}[\mathit{betty}{:}\mathit{tog}](\mathrm{K}_{\mathit{alice}}\mathit{light} \wedge \mathrm{K}_{\mathit{betty}}\mathit{light})$. In words, it means that after the parallel execution of Alice telling whether the light bulb is on or off to Betty

(TAU)	All instances of CPL tautologies	
(K)	$(\mathrm{K}_i\varphi \wedge \mathrm{K}_i(\varphi \to \psi)) \to \mathrm{K}_i\psi$	(deductive closure for knowledge)
(T)	$\mathrm{K}_i\varphi \to \varphi$	(truth)
(4)	$\mathrm{K}_i\varphi \to \mathrm{K}_i\mathrm{K}_i\varphi$	(positive introspection)
(5)	$\neg\mathrm{K}_i\varphi \to \mathrm{K}_i\neg\mathrm{K}_i\varphi$	(negative introspection)
(KA)	$([\delta\|_G]\varphi \wedge [\delta\|_G](\varphi \to \psi)) \to [\delta\|_G]\psi$	(deductive closure for action)
(A)	$\bigvee_{\delta\in\Delta} \neg[\delta\|_G]\bot$	(activity)
(DA)	$\bigwedge_{\delta'\in\Delta} [\delta\|_G \cup \delta'\|_{\mathbb{A}\setminus G}]\varphi) \to [\delta\|_G]\varphi$	(deriving action)
(S)	$([\delta\|_G]\varphi \wedge [\delta'\|_H]\psi) \to [\delta\|_G \cup \delta'\|_H](\varphi \wedge \psi)$ (if $G \cap H = \emptyset$)	(superadditivity)
(PR)	$\mathrm{K}_i[\delta\|_G]\varphi \to [\delta\|_G]\mathrm{K}_i\varphi$	(perfect recall)
(RMP)	From φ and $\varphi \to \psi$ infer ψ	(modus ponens)
(RNK)	From φ infer $\mathrm{K}_i\varphi$	(knowledge necessitation)
(RNA)	From φ infer $[\delta\|_G]\varphi$	(action necessitation)

Table 4.4 – Axiom system of CEDL

and Betty doing nothing, Betty knows that after Betty toggling the switch, the two agents know that the light is on. In other words, it means that if Betty waits for the announcement of Alice, then she knows how to reach the state where the two agents know that the light is on. ■

We now turn our attention to some important CEDL validities, displayed in Table 4.4.

Theorem 4.1. *The principles in Table 4.4 are sound and complete with respect to the class of CEDL models.*

The proof of Theorem 4.4 is done by providing an equivalent alternative semantics for CEDL and then using correspondence theory (Blackburn et al. 2001; Sahlqvist 1975) with this alternative semantics.

First, for every CEDL model $M = \langle W, R, T, I\rangle$ we build an alternative model $M^* = \langle W, R, T^*, I\rangle$. That is, M^* is the same structure as M, but with a different transition function T^*, which is defined as follows:

- T^* is a function from the set of all partial joint actions to $W \times W$, where $\langle w, w'\rangle \in T^*(\delta|_G)$ if and only if there is $\delta' \in \Delta$ such that $\langle w, w'\rangle \in T(\delta|_G \cup \delta'|_{\mathbb{A}\setminus G})$.

Alternative pointed models are tuples of the form $\langle M^*, w\rangle$, where M^* is as defined above and $w \in W$. The alternative satisfaction relation, $\vDash^*$, is the same as $\vDash$ for Boolean

operators and for operators K_i plus:

$$M^*, w \vDash^* [\delta|_G] \qquad \text{iff} \qquad \text{for all } w' \in T^*_{\delta|_G}(w) \text{ we have } M^*, w \vDash^* \varphi$$

Validity is defined as usual.

Second, let $\varphi \in \mathcal{L}_{\mathrm{CEDL}}$, we show by an induction on the structure of φ that $\langle M, w\rangle \vDash \varphi$ if and only if $\langle M^*, w\rangle \vDash^* \varphi$.

Third, we show that alternative models satisfy the following constraints:

(C1') $$\bigcup_{\delta\in\Delta} T^*_{\delta|_G}(w) \neq \emptyset$$

(C2') $$(T^*_{\delta|_G} \circ R_i)(w) \subseteq (R_i \circ T^*_{\delta|_G})(w)$$

(C3) $$T^*_{\delta|_G}(w) \subseteq \bigcup_{\delta'\in\Delta} T^*_{\delta|_G \cup \delta'|_{\mathbb{A}\setminus G}}(w)$$

(C4) $$T^*_{\delta|_{G\cup H}} \subseteq T^*_{\delta|_G}(w) \cap T^*_{\delta|_H}(w)$$

The first two are enforced by constraints (C1) and (C2) on CEDL models, respectively. The other two are enforced by the construction of T^*.

Fourth, it is easy to see that all axioms in Table 4.4 are Sahlqvist's formulas (Blackburn et al. 2001; Sahlqvist 1975). Then, by using the substitution algorithm, we obtain that Axioms (T), (4) and (5) correspond to reflexivity, transitivity and euclidicity of relations R_i, respectively, and Axioms (A), (DA), (S) and (PR) correspond to Constraints C1', C3, C4 and C2', respectively. Therefore, it follows from the Correspondence Theorem (Blackburn et al. 2001; Sahlqvist 1975) that the principles in Table 4.4 are sound and complete with respect to the class of alternative models. Since this semantics is equivalent to the semantics given earlier, then it is also sound and complete with respect to the class of CEDL models.

The axiom system in Table 4.4 reveals one additional assumption that is implicit in CEDL models. This assumption corresponds to Axiom (S), which is called 'superadditivity'. It stipulates that if group G obtains outcome φ by acting as determined by the partial joint action δ and H obtains outcome ψ by acting as determined by the partial joint action δ', then the group of agents $G \cup H$ obtains outcome $\varphi \wedge \psi$ by acting as determined by the union of their partial joint actions. In particular, this implies that the bigger the group, the more it can achieve. This seems to be an intuitive property.

4.2.3 Group Knowledge

Because we aim at formalising collective responsibility, we have to provide a formalisation of 'group knowledge'. Unfortunately, there is no consensus in the literature of what group knowledge means (see, e.g., the discussion in (Goldman and Blanchard 2018)). Here, we choose to use the notion of 'distributed knowledge', found, e.g., in (Fagin et al. 1995). In the language, we replace operators K_i by the more general K_G, which semantics is formally defined as follows. Let $G \neq \emptyset$:

$$M, w \vDash \mathrm{K}_G\varphi \qquad \text{iff} \qquad M, w' \vDash \varphi \text{ for all } w' \in \bigcap_{i\in G} R(i)(w)$$

Distributed knowledge approximately describes the knowledge of someone who has complete knowledge of what each member of the community knows.

Proposition 4.2. *The following schemata are valid in CEDL with distributed knowledge:*

(K′)	$(\mathrm{K}_G\varphi \wedge \mathrm{K}_G(\varphi \to \psi)) \to \mathrm{K}_G\psi$	*(deductive closure for knowledge)*
(T′)	$\mathrm{K}_G\varphi \to \varphi$	*(truth)*
(4′)	$\mathrm{K}_G\varphi \to \mathrm{K}_G\mathrm{K}_G\varphi$	*(positive introspection)*
(5′)	$\neg\mathrm{K}_G\varphi \to \mathrm{K}_G\neg\mathrm{K}_G\varphi$	*(negative introspection)*
(KS)	$(\mathrm{K}_{G_1}\varphi_1 \wedge \mathrm{K}_{G_2}\varphi_2) \to \mathrm{K}_{G_1\cup G_2}(\varphi_1 \wedge \varphi_2)$ *(if $G \cap H = \emptyset$)*	*(knowledge superadditivity)*
(PR′)	$\mathrm{K}_G[\delta|_{\mathbb{A}}]\varphi \to [\delta|_{\mathbb{A}}]\mathrm{K}_G\varphi$	*(perfect recall)*

Axioms (K′) to (5′) are, again, shown using correspondence theory. Axiom (PR′) is valid because (C2) is preserved for groups G, by the definition of K_G. And (KS) is valid because $\bigcap_{i\in G_1\cup G_2} R(i) \subseteq \bigcap_{i\in G_1} R(i)$.

Note that using axioms (KS) and (K′), one can derive both: $\mathrm{K}_{G_1}\varphi \to \mathrm{K}_{G_1\cup G_2}\varphi$ and $(\mathrm{K}_{G_1}\varphi_1 \wedge \mathrm{K}_{G_2}(\varphi_1 \to \varphi_2)) \to \mathrm{K}_{G_1\cup G_2}\varphi_2$. However, whether these principles constitute a complete axiom system for CEDL with distributed knowledge, is left as an open question.

4.2.4 Ability and Knowing How Ability

Our aim in this section is to define operators to express agent ability. The first operator we define here expresses that 'by executing $\delta|_G$, the group of agents G ensures an outcome satisfying φ in the next step'. That is, formulas of the form $\mathrm{E}_{\delta|_G}\varphi$ should mean that G can execute $\delta|_G$ (or, simply, $\delta|_G$ is executable) and that it necessarily leads to an outcome satisfying φ. It therefore amounts to the following abbreviation:

$$\mathrm{E}_{\delta|_G}\varphi \stackrel{\text{def}}{=} \neg[\delta|_G]\bot \wedge [\delta|_G]\varphi$$

Sometimes, we will need to express that some group of agents ensure an outcome φ by executing a sequence of actions $\delta_1|_G; \ldots; \delta_2|_G$. This amounts to a similar abbreviation, which generalizes the previous one:

$$\mathrm{E}_{\delta_1|_G;\ldots;\delta_n|_G}\varphi \stackrel{\text{def}}{=} \neg[\delta_1|_G]\ldots[\delta_n|_G]\bot \wedge [\delta_1|_G]\ldots[\delta_n|_G]\varphi$$

The second operator defined here expresses that 'the group of agents G is able to ensure that φ is true in the next step'. That is, formulas of the form $\langle\!\langle G\rangle\!\rangle\varphi$ should mean that G has an available action such that G can execute and which ensures an outcome satisfying φ. Thus, the latter is given by:

$$\langle\!\langle G\rangle\!\rangle\varphi \stackrel{\text{def}}{=} \bigvee_{\delta\in\Delta} \mathrm{E}_{\delta|_G}\varphi$$

We note that this is a well-formed formula, because Δ is finite, since $\mathbb{T}$ is finite.

Example 4.5 (Light Bulb and Light Switch (revisited))**.** To exemplify operators $\langle\!\langle G \rangle\!\rangle$, we reuse the scenario of Example 4.4 and the model in Figure 4.2.

- $\langle M, w_0 \rangle \vDash \langle\!\langle \{betty\} \rangle\!\rangle light$.
  In words, Betty is able to ensure that the light is on in the next step. This is true because there is an action available for Betty such that she can execute and which ensures an outcome satisfying φ. This action is *tog*. Let us see how it works formally. The claim is true if and only if $\langle M, w_0 \rangle \vDash \bigvee_{\delta \in \Delta} \mathrm{E}_{\delta|_G} light$, which is true if $\langle M, w_0 \rangle \vDash \mathrm{E}_{betty:tog} light$, The latter is true if and only if $\langle M, w_0 \rangle \vDash \neg[betty{:}tog]\bot$ and $\langle M, w_0 \rangle \vDash [betty{:}tog] light$. Both are indeed the case, as we have already seen in Example 4.4.

- $\langle M, w_0 \rangle \vDash \neg\langle\!\langle \{alice\} \rangle\!\rangle \mathrm{K}_{betty} \neg light$.
  In words, it is not the case that Alice is able to ensure that Betty knows that the light is off in one step. This is the case because there is no action available for Alice that ensures that Betty knows that the light is on on the next step. For instance, Betty can always toggle the switch, which leads to the situation where the light is off. Formally, the two options available for Alice do not lead, necessary, to a situation where the light is off. That is, we have that $\langle M, w_0 \rangle \vDash \neg[alice{:}tell]\mathrm{K}_{betty} \neg light$, as seen before, and also $\langle M, w_0 \rangle \vDash \neg[alice{:}none]\mathrm{K}_{betty} \neg light$.

- $\langle M, w_0 \rangle \vDash \langle\!\langle \{alice\} \rangle\!\rangle \langle\!\langle \{betty\} \rangle\!\rangle \mathrm{K}_{betty} light$. In words, Alice is able to ensure that after one step Betty is able to ensure after one more more step that she knows that the light is on. In the first step, the action available for Alice that leads to that result is *tell*. The reader can verify that, if Alice executes *tell* an Betty does noting, then after that she knows that the light is off, and then she only has to toggle the switch to ensure that the light is on after one more step. On the other hand, if Alice executes *tell* and Betty toggles the switch, then after that she knows that the light is on, and then she only has to do nothing to ensure that the light is on after one more step. So, in any case, Betty will have an action that leads to a state where she knows that the light is on.

- $\langle M, w_0 \rangle \vDash \langle\!\langle \{alice, betty\} \rangle\!\rangle light \wedge \langle\!\langle \{alice, betty\} \rangle\!\rangle \neg light$.
  In words, the two agents together are able to ensure that the light is on and they are able to ensure that the light is off. This is true because the agents may chose to execute, e.g., *alice*:*none*, *betty*:*tog* to ensure that the light is on after one step And also, they can chose to execute, e.g., *alice*:*none*, *betty*:*none* to ensure that the light is off after one step. This shows that in this scenario, the agents are able to control the state of propositional variable *light*.

- $\langle M, w_0 \rangle \vDash \neg\langle\!\langle \emptyset \rangle\!\rangle light \wedge \neg\langle\!\langle \emptyset \rangle\!\rangle \neg light$.
  We note that this is a well-formed formula, because $\emptyset \subseteq \mathbb{A}$. Thus, we can follow the definition of $\langle\!\langle \emptyset \rangle\!\rangle$, to verify that, for instance, the first conjunct $\neg\langle\!\langle \emptyset \rangle\!\rangle light$ is true at w_0. To do so, it is enough to verify that for all $\delta \in \Delta$, the formula $[\delta|_\emptyset] light$ is false at w_0. The latter is the case if and only if for all $\delta \in \Delta$ there is $\delta' \in \Delta$ such that the formula $[\delta|_\emptyset \cup \delta'_{\{alice,betty\}}] light$ is false at w_0. And the latter is indeed

the case for, e.g., $\delta' = \mathit{alice}{:}\mathit{none}, \mathit{betty}{:}\mathit{none}$. To verify that the second conjunct is also true at w_0, we do the same reasoning but using $\delta' = \mathit{alice}{:}\mathit{none}, \mathit{betty}{:}\mathit{tog}$. ■

At this point, the reader may be wondering what interpretation should be given to $\langle\!\langle \emptyset \rangle\!\rangle$. Formulas of the form $\langle\!\langle \emptyset \rangle\!\rangle \varphi$ mean 'after the execution of any joint action, φ is true', which is equivalent to 'φ is true in the next step, in spite of what the agents do' or even, 'it is necessary the case that φ true in the next step'. This interpretation of $\langle\!\langle \emptyset \rangle\!\rangle$ is similar as given to its homonym in Coalition Logic (CL) and Alternating-time Temporal Logic (ATL) (Alur et al. 2002). In fact, for all $G \subseteq \mathbb{A}$, the interpretation given to $\langle\!\langle G \rangle\!\rangle$ is similar to that in CL and ATL, but with some technical differences. Our $\langle\!\langle G \rangle\!\rangle$ is the fusion of ATL operators $\langle\!\langle G \rangle\!\rangle$ and $\mathbf{X}$ (where the latter means "next"). That is, our formulas of the form $\langle\!\langle G \rangle\!\rangle \varphi$ correspond to ATL formulas of the form $\langle\!\langle G \rangle\!\rangle \mathbf{X} \varphi$. It turns out that our operator $\langle\!\langle G \rangle\!\rangle$ validates some of the axioms and inference rule of ATL operators $\langle\!\langle G \rangle\!\rangle \mathbf{X}$ (as found, e.g., in (Goranko and van Drimmelen 2006)).

Proposition 4.3. *The following schemas and rule of inference are valid in CEDL:*

1. $\neg\langle\!\langle G \rangle\!\rangle \bot$
2. $\langle\!\langle G \rangle\!\rangle \top$
3. $(\langle\!\langle G \rangle\!\rangle \varphi \wedge \langle\!\langle H \rangle\!\rangle \psi) \to \langle\!\langle G \cup H \rangle\!\rangle (\varphi \wedge \psi)$ *(if $G \cap H = \emptyset$)*
4. *From $\varphi \to \psi$ infer $\langle\!\langle G \rangle\!\rangle \varphi \to \langle\!\langle G \rangle\!\rangle \psi$*

It is easy to see why Proposition 4.3 holds:

1. Suppose that $\langle M, w \rangle \vDash \langle\!\langle G \rangle\!\rangle \bot$, for some arbitrary pointed model $\langle M, w \rangle$. Then, there is a δ such that $\langle M, w \rangle \vDash \neg[\delta|_G]\bot \wedge [\delta|_G]\bot$, which is a contradiction.
2. From Axiom (A), $\vDash \exists_{\delta \in \Delta} \neg[\delta|_G]\bot$ for all $G \subseteq \mathbb{A}$. Since also $\vDash [\delta|_G]\top$ for all $\delta \in \Delta$ and all $G \subseteq \mathbb{A}$, we have $\vDash \exists_{\delta \in \Delta}(\neg[\delta|_G]\bot \wedge [\delta|_G]\top)$.
3. First, note that $\langle\!\langle G \rangle\!\rangle \varphi \wedge \langle\!\langle H \rangle\!\rangle \psi$ is equivalent to the formula: $\exists_{\delta \in \Delta}(\neg[\delta|_G]\bot \wedge [\delta|_G]\varphi) \wedge \exists_{\delta' \in \Delta}(\neg[\delta'|_H]\bot \wedge [\delta'|_H]\psi)$. Also note that (a) $\vDash \neg[\delta|_G \cup \delta'|_H]\bot$ (by Axiom (A)), and also (b) $\vDash ([\delta|_G]\varphi \wedge [\delta'|_H]\psi) \to [\delta|_G \cup \delta'|_H](\varphi \wedge \psi)$ (by Axiom (S)). Putting (a) and (b) together we have $\vDash \langle\!\langle G \rangle\!\rangle \varphi \wedge \langle\!\langle H \rangle\!\rangle \varphi \to \langle\!\langle G \cup H \rangle\!\rangle (\varphi \wedge \psi)$.
4. From Axiom (A), (a) $\vDash \exists_{\delta \in \Delta} \neg[\delta|_G]\bot$ for all G. And from $\vDash \varphi \to \psi$ we infer $\vDash [\delta|_G](\varphi \to \psi)$ (by Rule (RNA)), and then (b) $\vDash [\delta|_G]\varphi \to [\delta|_G]\psi$ (by Axiom (KA) and Rule (RMP)). Then, (a) and (b) together imply $\vDash \langle\!\langle G \rangle\!\rangle \varphi \to \langle\!\langle G \rangle\!\rangle \psi$.

We remark that the formula $\langle\!\langle \mathbb{A} \rangle\!\rangle \varphi \to \neg\langle\!\langle \emptyset \rangle\!\rangle \neg\varphi$ is valid. Indeed, we can derive it quite easily using Axioms (S) and (A). This formula means that if the whole set of agents is able to ensure that φ is true after one step, then it is not the case that φ is necessarily false after one step. However, the converse is not the case. That is, $\neg\langle\!\langle \emptyset \rangle\!\rangle \neg\varphi \to \langle\!\langle \mathbb{A} \rangle\!\rangle \varphi$ is not valid in CEDL. In words, just because φ is not necessarily false after one step, it does not mean that the whole set of agents are able to ensure it. It happens because we do not assume 'joint determinism' in CEDL, i.e., a joint action may have more than

one possible outcome. It contrasts with ATL, where the formula $\neg\langle\!\langle\emptyset\rangle\!\rangle\mathbf{X}\neg\varphi \rightarrow \langle\!\langle\mathbb{A}\rangle\!\rangle\mathbf{X}\varphi$ is valid. This is not the case here though.[1]

Theorem 4.3 shows that operators $\langle\!\langle G\rangle\!\rangle$ are similar to their CL and ATL homonyms. This means that these operator presents well-known, and desired, properties of an operator supposed to model ability of agents. It is very important for our framework, since we will base our subsequent definitions in the intuition behind the concept of ability.

Nonetheless, this operator has a "problem". It has been argued several times (Ågotnes and van Ditmarsch 2008; Broersen et al. 2007; Jamroga and Ågotnes 2007; Jamroga and van der Hoek 2004) that, when a logic of this kind is also able to express that agents' knowledge is incomplete, the operator just defined is not completely adequate. The same issue rises in our logic, and can be explained using the Light Bulb and Light Switch scenario again (Examples 4.4 and 4.5): let us recall that Betty is able to ensure that the light is on after one step. That is, $\langle\!\langle\{betty\}\rangle\!\rangle light$ is true at w_0. In fact, this formula is true at w_1 too. The latter means that she knows it, i.e., $\langle M, w_0\rangle \vDash \mathrm{K}_{betty}\langle\!\langle\{betty\}\rangle\!\rangle light$. Then, it may seem counterintuitive that Betty does not know that the action *tog* ensures *light* in w_0, i.e., $\langle M, w_0\rangle \vDash \neg\mathrm{K}_{betty}\mathrm{E}_{betty:tog} light$. However, the reader may verify that it is indeed the case. This means that although Betty knows that she is able to ensure that the light is on after one step, she does not know what she must do to ensure it! In game theory (Osborne and Rubinstein 1994) we say that an agent has a 'non-uniform strategy' for a given goal whenever:

> for every state that the agent cannot distinguish from the current state, there is a strategy whose execution leads to the goal;

and we say that an agent has a 'uniform strategy' for a given goal whenever:

> there is a strategy such that for every state that the agent cannot distinguish from the current one, its execution leads to the goal.

In our example above, Betty has a 'non-uniform strategy' for the outcome where the light is on. The formula $\mathrm{K}_{betty}\langle\!\langle betty\rangle\!\rangle light$ expresses it. However, sometimes we would like to write a formula expressing that an agent has (or does not have) a uniform strategy for a given goal. In the language of coalition logic and ATL, the latter is not possible. But this is possible in CEDL. The formula $\bigvee_{\delta\in\Delta}\mathrm{K}_i\mathrm{E}_{\delta|_i} light$ (which is false at w_0) expresses that Betty has a uniform strategy to obtain the goal *light*. In other words, it expresses that Betty knows how to ensure that the light is on after one step. Here, we call this a 'knowing how ability'. To be able to express it more succinctly, we define operators H_G, as follows:

$$\mathrm{H}_G\varphi \stackrel{\text{def}}{=} \bigvee_{\delta\in\Delta}\mathrm{K}_G\mathrm{E}_{\delta|_G}\varphi$$

The knowing how operator has some of the properties of the ability operator. For instance, we have the following.

Proposition 4.4. *The following scheme and rule of inference are valid in CEDL:*

[1] This constitutes a difference from the logic proposed in (de Lima et al. 2010b). There, joint determinism is taken as an assumption.

1. $\neg \mathrm{H}_G \bot$
2. $(\mathrm{H}_{G_1}\varphi_1 \wedge \mathrm{H}_{G_2}\varphi_2) \rightarrow \mathrm{H}_{G_1 \cup G_2}(\varphi_1 \wedge \varphi_2)$ *(if* $G \cap H = \emptyset$*)*
3. *From* $\varphi \rightarrow \psi$ *infer* $\mathrm{H}_G\varphi \rightarrow \mathrm{H}_G\psi$

But not all properties are the same. For instance, $\langle\!\langle G \rangle\!\rangle \top$ is valid, while $\mathrm{H}_G \top$ is not. Validity of the former means that there is always some action available for G that is executable, because of Axiom A. The latter is not valid because agent G does not always know which action it is.

4.2.5 Obligations

In this section, we define an operator expressing agents obligations, as in deontic logic (Meyer and Wieringa 1993). We do so by adapting the simple, yet effective, idea of d'Altan et al. (1996) to our framework. The latter developed further the ideas of Meyer (1988), introducing both static and dynamic obligations in PDL.

From now on, we assume that the set of propositional variables of our language is $\mathbb{P} \cup \mathit{Vio}$, where $\mathit{Vio} = \{\mathit{vio}_G \mid G \in 2^{\mathbb{A}} \setminus \emptyset\}$. That is, for each agent, we introduce a new atomic formula vio_G, that has the special meaning: 'the group of agents G is in violation'. In addition, we require that our models now be quadruples of the form $\langle W, R, T, I\rangle$, where W, R and T are defined as before but the domain of I is extended to the new set of propositional variables, i.e., $I : (\mathbb{P} \cup \mathit{Vio}) \rightarrow 2^W$.

In (d'Altan et al. 1996), formulas of the form $\mathrm{O}_G\varphi$ mean 'it is obligatory for G that φ is true'. The idea is that vio_G flags the states of the model that consists of a violation state for G. In other words, vio_G simply means 'G is in violation'. Thus, we define one obligation operator for each group of agents $G \in 2^{\mathbb{A}} \setminus \emptyset$ with the following abbreviation:

$$\mathrm{O}_G\varphi \stackrel{\text{def}}{=} \neg\varphi \rightarrow \mathit{vio}_G$$

Obligations satisfy some interesting properties.

Proposition 4.5.

1. $\vDash \mathrm{O}_G \top$
2. $\vDash \mathrm{O}_G(\varphi \wedge \psi) \leftrightarrow (\mathrm{O}_G\varphi \wedge \mathrm{O}_G\psi)$
3. *If* $\vDash \varphi$ *then* $\vDash \mathrm{O}_G\varphi$

We note that agents do not necessarily know their obligations, i.e., $\mathrm{O}_G\varphi$ does not imply $\mathrm{K}_G\mathrm{O}_G\varphi$. Also, our approach permits models where violations are unavoidable. For instance, we have that $\mathit{vio}_G \rightarrow \mathrm{O}_G\bot$ is valid, which also means that we do not impose Axiom D (i.e., $\neg\mathrm{O}_G\bot$), which is present in some deontic systems. In our view it just means that dilemmas are possible (cf. van Fraassen (1973)). As in the classical Sartre's example, one can have the obligation to stay at home to look after an elderly mother and, at the same time, to have obligation to join the resistance movement to fight the Nazis.

Now, still following d'Altan et al. (1996) and Meyer (1988), we can also define dynamic obligations in our framework, by using the dynamic operator available in our language, i.e., the operator [ ]. We therefore define:

$$\mathbf{F}_G(\delta|_G) \stackrel{\text{def}}{=} [\delta|_G]vio_G$$
$$\mathbf{F}_G(\delta^1|_G;\ldots;\delta^n|_G) \stackrel{\text{def}}{=} \mathbf{F}_G(\delta|_G) \vee [\delta^1|_G]\mathbf{F}_G(\delta^2|_G;\ldots;\delta^n|_G)$$

A formula of the form $\mathbf{F}_G(\delta|_G)$ literally means that, 'G will necessarily be in a violation state after every possible occurrence of $\delta|_G$.' It should be read as 'action $\delta|_G$ is forbidden for G'. Permitted actions are also defined by an abbreviation:

$$\mathbf{P}_G(\delta^1|_G;\ldots;\delta^n|_G) \stackrel{\text{def}}{=} \neg\mathbf{F}_G(\delta^1|_G;\ldots;\delta^n|_G)$$

That is, action $\delta|_G$ is permitted for G if and only if $\delta|_G$ is not forbidden for G.

Our operator O has an important difference compared to the one defined by Meyer (1988). Meyer's obligation is abbreviated by a master modality $\Box$,[2] while ours is not. Therefore, our obligations are "immediate" ones, instead of Meyer's "global" ones. That is, $\mathrm{O}_G\varphi$ means that φ is obligatory only now. We think that this kind of obligation makes sense, too. It may even make more sense than Meyer's obligations, since in real life, some rules and norms may expire, or even be subject of change.

4.3 Responsibility

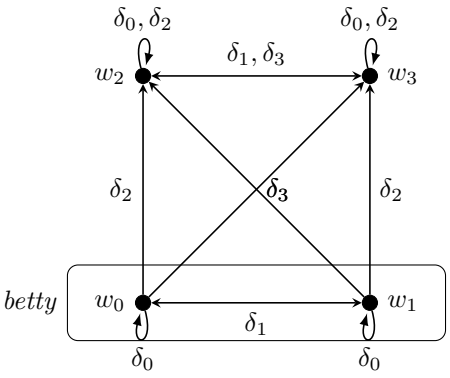
With all the necessary ingredients in hands, we are now able to develop our formal theory of responsibility. We start with forward-looking responsibility, in Subsection 4.3.1, and then we deal with the two different kinds of backward-looking responsibilities, in Subsection 4.3.2. Subsection 4.3.3 establishes the relation between these two kinds of responsibility.

4.3.1 Forward-looking Responsibility

In this section, we formalize one kind of forward-looking responsibility, namely, the 'obligation to see to it that'. The aim here is to augment our logic with operators $\mathcal{R}$, where formulas of the form $\mathcal{R}^n_G\varphi$ are to be read as 'it is obligatory for G that G sees to it that φ is true after n steps'. But, before showing its definition, let us recall part of the informal discussion drawn in van de Poel et al. 2015, Chapter 1, where Goodin's ideas are used to define this meaning of responsibility. The argument advanced is the following quote from Goodin 1995, p. 83:

> The standard form of responsibility is that A *see to it that* X. It is not enough that X occurs. A must have "seen to it" that X occurs. "Seeing to it that X" requires, minimally: that A satisfy himself that there is some

[2] In the present context a master modality $\Box$ would be an operator satisfying: (i) $\Box\varphi \to [\delta|_G]\varphi$, for all $\delta|_G$; (ii) $\Box\varphi \to \varphi$; and (iii) $\Box\varphi \to \Box\Box\varphi$.



$I(light) = \{w_1, w_3\}$
$I(vio_{alice}) = \{w_0, w_1, w_2\}$
$I(vio_{betty}) = \{w_0, w_1, w_2\}$
$\delta_0 = alice{:}none, betty{:}none$
$\delta_1 = alice{:}none, betty{:}tog$
$\delta_2 = alice{:}tell, betty{:}none$
$\delta_3 = alice{:}tell, betty{:}tog$

Figure 4.3 – A CEDL model for the Light Bulb and Light Switch scenario where w_0 and w_2 are violation states. (As before, the rectangles represent Betty's knowledge. To simplify the picture, Alice's knowledge is not represented. She has complete knowledge of the scenario, i.e., $R_{alice}(w) = \{w\}$ for all $w \in W$.)

> process (mechanism or activity) at work whereby X will brought about; that A check from time to time to make sure that process is still at work, and is performing as expected; and that A take steps as necessary to alter or replace process that no longer seem likely to bring about X.

This is a complex definition. It is unlikely that we would be able to capture all its details in our framework. Nonetheless, we approximate it by the following inductive definition:

$$\mathcal{R}^0_G\varphi \stackrel{\text{def}}{=} \mathrm{O}_G\varphi \tag{4.1}$$

$$\mathcal{R}^{n+1}_G\varphi \stackrel{\text{def}}{=} \mathrm{O}_G\mathrm{H}^{n+1}_G\varphi \wedge \langle\!\langle\emptyset\rangle\!\rangle\mathcal{R}^n_G\varphi \qquad \text{(for } n > 0\text{)} \tag{4.2}$$

where H^n_G stands for a sequence of n operators H_G, i.e., $\mathrm{H}^0_G\varphi \stackrel{\text{def}}{=} \varphi$ and $\mathrm{H}^n_G\varphi \stackrel{\text{def}}{=} \mathrm{H}_G\mathrm{H}^{n-1}_G\varphi$.

The formula $\mathcal{R}^0_G\varphi$ is equivalent to $\mathrm{O}_G\varphi$. That is, $\mathcal{R}^0_G\varphi$ is true at some world w if and only if, it is necessary the case that, $\neg\varphi$ implies a violation for G. But formulas $\mathcal{R}^n_G\varphi$, for $n > 1$, are more than mere obligations that φ. They also capture the "supervisory nature" of this kind of responsibility. For it is also necessary the case that, after $n-1$ steps, $\neg\mathrm{H}_G\varphi$ implies a violation for G. This means that, to avoid a violation, group G must find a way to be in the position to know how to ensure φ after one step. And analogously for $n-2$, and so on until $n = 1$.

Example 4.6 (Light Bulb and Light Switch with Violations)**.** Let us now see our operator $\mathcal{R}$ in action using a variation of Example 4.4. The only difference in this variation is that worlds w_0, w_1 and w_2 are violation states for Alice and also for Betty. Such scenario is depicted in Figure 4.3.

- $\langle M, w_0\rangle \vDash \mathcal{R}^1_{alice}\mathit{light}$ and also $\langle M, w_0\rangle \vDash \mathcal{R}^1_{betty}\mathit{light}$
  In words, it is obligatory for Alice to see to it that the light is on after one step, and the same for Betty. We verify only Alice's case, since Betty's one is analogous. It is the case if and only if (1) $\langle M, w_0\rangle \vDash \mathrm{O}_{alice}\mathrm{H}_{alice}\mathit{light}$ and (2) $\langle M, w_0\rangle \vDash \langle\!\langle\emptyset\rangle\!\rangle\mathcal{R}^0_{alice}\mathit{light}$. We have (1) because $\langle M, w_0\rangle \vDash \mathit{vio}_{alice}$. And we have (2) because, for every w' in the model, we have $M, w' \vDash \neg\mathit{light} \to \mathit{vio}_{alice}$.

- $\langle M, w_0\rangle \vDash \mathcal{R}^2_{alice}\mathit{light}$
  In words, it is obligatory for Alice to see to it that the light is on after two steps. To show it, we have to show that (1) $\langle M, w_0\rangle \vDash \mathrm{O}_{alice}\mathrm{H}_{alice}\mathrm{H}_{alice}\mathit{light}$ and (2) $\langle M, w_0\rangle \vDash \langle\!\langle\emptyset\rangle\!\rangle\mathcal{R}^1_{alice}\mathit{light}$. To show (1) we have to show that $\neg\mathrm{H}_{alice}\mathrm{H}_{alice}\mathit{light} \to \mathit{vio}_{alice}$ is true at w_0. The reader can verify that this is indeed the case. To show (2) we have to show that for all $w \in W$ we have $M, w \vDash \mathcal{R}^1_{alice}\mathit{light}$. It can be done in the same way as for w_0, which is done in the previous item.

We also note that that $\langle M, w_0\rangle \vDash \mathcal{R}^2_{betty}\mathit{light}$, and invite the reader to do the exercise of showing it. ∎

It may worth to explain why we do not use the classical notion of 'seeing to it that' (henceforth stit) proposed in (Belnap et al. 2001). Belnap et al.'s stit can be defined in our framework as follows. The sentence 'by executing $\delta|_{\{i\}}$, i sees to it that φ after one step' is expressed by the formula $\neg[\delta|_{\{i\}}]\bot \wedge [\delta|_{\{i\}}]\varphi \wedge \neg\langle\!\langle\emptyset\rangle\!\rangle\varphi$. Note that it amounts to ours $\mathrm{E}_{\delta|_{\{i\}}}\varphi$ plus the conjunct $\neg\langle\!\langle\emptyset\rangle\!\rangle\varphi$. This conjunct is there because, for Belnap et al., it is not enough that φ is true after one step. It must be the agent i that influences the course of things to ensure that φ is true after one step. It can be argued that our operator E does not capture this latter meaning. For example, the formula $\mathrm{E}_{\delta|_G}\top$ is valid in CEDL, i.e., it is true at every world of every model. However, Belnap et al.'s stit is incompatible with the meaning of responsibility we are trying to formalize here. We show it using an informal example. Suppose that a teacher has the obligation to see to it that, after his class, the door of the classroom will be closed. Also suppose that the door of the classroom is connected to an automatic system that closes it exactly after his class finishes. Some minutes before finishing his class, the teacher has some options: (a) finish the class on time and then let the door be closed automatically by the system; (b) finish the class a bit earlier and then close the door himself; and (c) disable the automatic system an then close the door himself. It seems to us that in Belnap et al.'s terms, only (b) and (c) fulfill the teacher's obligation, while we think that all the three options fulfill the teacher's obligation. On the other hand, recall that, as explained earlier, the obligation to see to it that φ does not require that φ is achieved by an action of the teacher. But it requires some supervision. Therefore, in our example, the obligation to see to it that the door will be closed implies that the agent supervises the process by himself. That is, if the teacher chooses (a), then the teacher must be sure that the system is working properly or at least verify if the system closed the door after the class.

4.3.2 Backward-looking Responsibility

In this section, we formalize two kinds of individual backward-looking responsibility, 'accountability' and 'blameworthiness'. The aim here is to augment the logic with two operators, $\mathcal{A}$ and $\mathcal{B}$. Formulas of the form $\mathcal{A}_{\delta|_i}\varphi$ is to be read as 'after executing δ, agent i is accountable for φ' and formulas of the form $\mathcal{B}_{\delta|_i}\varphi$ is to be read as 'after executing δ, agent i is blamed for φ'.

Accountability

Let us start by recalling the definition of 'accountability' given in van de Poel et al. 2015, Chapter 1:

> ... an agent A is accountable for X if A has the capacity to act responsibly (has moral agency), is somehow causally connected to the outcome X (by an action or omission) and there is a reasonable suspicion that agent A did somehow something wrong.

The first condition for holding an agent accountable, 'moral agency', is a tacit assumption in our framework. In fact, our logical framework also assumes that all agents are rational and that they are perfect reasoners. This, for instance, means that agents can foresee all the logical consequences of the facts they know. Again, these assumptions may be seen as very restrictive ones, but, in a logical framework like ours, the relaxation of such assumptions would make it much more difficult (if not impossible) to derive some interesting properties.

The other two conditions for holding an agent accountable are causality and wrong-doing. We just make it precise that these two conditions must be related to each other, in the sense that the "something wrong" did by the agent must be what is "somehow causally connected to the outcome" for which the agent is accountable. So, first, we consider that agent i is causally connected to a given consequence φ whenever i ensures that φ is true and the action that ensures φ is wrong. And we consider that an action is wrong whenever it does not ensure a non-violation state. For the latter, it either means that a violation state is achieved by the execution of the action or, if the violation state is not achieved, it means that the agent did not take enough care in order to avoid it, what we also consider as a "wrong-doing". Putting everything together we have the following definition for accountability (which is already generalised to groups of agents):

$$\mathcal{A}_{\delta_1|_G;\ldots;\delta_n|_G}\varphi \stackrel{\text{def}}{=} \mathrm{E}_{\delta_1|_G;\ldots;\delta_n|_G}\varphi \wedge \neg\mathrm{E}_{\delta_1|_G;\ldots;\delta_n|_G}\neg \mathit{vio}_G$$

The latter definition stipulates that, after executing the sequence of actions $\delta_1;\ldots;\delta_2$, group G is accountable for φ if and only if, by executing this sequence, G ensures that φ is true and also, by executing this sequence, G does not ensure a non-violation state.

Example 4.7 (Light Bulb and Light Switch with Violations (revisited))**.** To exemplify operator $\mathcal{A}$, we reuse the scenario of Example 4.6 and the model in Figure 4.3.

- $\langle M, w_0\rangle \vDash \neg\mathcal{A}_{\delta_0|_{alice}}\neg \mathit{light}$.
  In words, it is not the case that, after executing action *none*, Alice is accountable for the fact that the light is off. It is true if and only if $\neg\mathrm{E}_{\delta_0|_{alice}}\neg \mathit{light} \vee$

$\mathrm{E}_{\delta_0|_{alice}}\neg vio_{alice}$. And it is true because the first disjunct is true. To see it, note that $\langle M, w_0\rangle \vDash \neg[\delta_0|_{alice}]\neg light$.

- $\langle M, w_0\rangle \vDash \mathcal{A}_{\delta_0|_{betty}}\neg light$.
  In words, Betty is accountable for the fact that, after executing action *none*, the light is off. It is true if and only if $\mathrm{E}_{\delta_0|_{betty}}\neg light \wedge \neg\mathrm{E}_{\delta_0|_{betty}}\neg vio_{betty}$. Note that both conjuncts are true in the model of Figure 4.3.

- $\langle M, w_0\rangle \vDash \neg\mathcal{A}_{\delta_0|_{alice};\delta_0|_{alice}}\neg light$.
  In words, it is not the case that, after executing action *none* twice, Alice is accountable for the fact that the light is off. To show it, we use the same reasoning as for the first item again, but now twice.

We may say that, in this example, operator $\mathcal{A}$ works as expected. Note that Alice, who does not control the light switch cannot be held accountable for the fact that the light is off. On the other hand, Betty, who does control the light switch, can be held accountable for the fact that the light is off. But, we will see in the next section that she cannot always be blamed for that, because she does not know the state of the light in the beginning. ■

Blameworthiness

The definition given in van de Poel et al. 2015, Chapter 1 for this meaning of responsibility is the following: agent i is blamed for consequence φ whenever i is accountable for φ and is not capable to give an account for it. Two accounts (excuses) are considered acceptable. The first one is ignorance: the agent i is excused (and thus is not blamed) for φ if i can show that it does not know that its behavior causes φ, or it does not know that its behavior is wrong. The second is compulsion: the agent i is excused for φ if i can show that no behavior that does not cause φ or that is not wrong is possible. For simplicity, we give the definition of blame in two steps (note the generalisation to groups of agents):

$$\mathcal{C}_{\delta|_G}\varphi \stackrel{\text{def}}{=} \mathrm{K}_G\mathrm{E}_{\delta|_G}\varphi \wedge \neg\langle\!\langle\emptyset\rangle\!\rangle\varphi$$

$$\mathcal{C}_{\delta_1|_G;\ldots;\delta_n|_G}\varphi \stackrel{\text{def}}{=} \mathrm{K}_G\mathrm{E}_{\delta_1|_G}\mathcal{C}_{\delta_2|_G;\ldots\delta_n|_G}\varphi \wedge \neg\langle\!\langle\emptyset\rangle\!\rangle\mathcal{C}_{\delta_2|_G;\ldots;\delta_n|_G}\varphi \qquad (\text{for } n>1)$$

$$\mathcal{B}_{\delta_1|_G;\ldots;\delta_n|_G}\varphi \stackrel{\text{def}}{=} \mathcal{C}_{\delta_1|_G;\ldots;\delta_n|_G}\varphi \wedge \neg\mathcal{C}_{\delta_1|_G;\ldots;\delta_n|_G}\neg vio_G$$

The definition of operator $\mathcal{B}$ is similar to that of $\mathcal{A}$, but with operator $\mathcal{C}$ in the place of ensure. The latter operator means more than just ensure. It could be seen as 'knowing causality'. That is, formula of the form $\mathcal{C}_{\delta|_G}\varphi$ mean 'by executing δ, group G knowingly causes φ'. The difference with ensuring φ is that the latter formula also means that the agent knows that the action ensures φ and, in addition, the conjunct $\neg\langle\!\langle\emptyset\rangle\!\rangle\varphi$ means that an option not necessarily leading to φ was possible. Therefore, the definition of formula $\mathcal{B}_{\delta|_G}\varphi$ stipulates that, not only the group is accountable for φ, but the group cannot provide acceptable excuses for it, which, by the defition given initially, means that the group is blamed for φ.

Example 4.8 (Light Bulb and Light Switch with Violations (re-revisited))**.** To exemplify operator $\mathcal{B}$, we reuse the scenario of Example 4.6 and the model in Figure 4.3.

- $\langle M, w_0\rangle \vDash \neg\mathcal{B}_{\delta_0|_{alice}}\neg \mathit{light}$.
  In words, it is not the case that, after executing action *none*, Alice is blamed for the fact that the light is off. It is true if and only if $\langle M, w_0\rangle \vDash \neg\mathcal{C}_{\delta_0|_{alice}}\neg \mathit{light} \vee \mathcal{C}_{\delta_0|_{alice}}\neg \mathit{vio}_{alice}$. And it is true because the first disjunct is true. To see it, note that $\langle M, w_0\rangle \vDash \neg \mathrm{K}_{alice}\mathrm{E}_{\delta_0|_{alice}}\neg \mathit{light}$, because $\langle M, w_0\rangle \vDash \neg \mathrm{K}_{alice}[\delta_0|_{alice}]\neg \mathit{light}$.

- $\langle M, w_0\rangle \vDash \neg\mathcal{B}_{\delta_0|_{betty}}\neg \mathit{light}$.
  In words, it is not the case that, after executing the action *none*, Betty is blamed for the fact that the light is off. It is true if and only if $\langle M, w_0\rangle \vDash \neg\mathcal{C}_{\delta_0|_{betty}}\neg \mathit{light} \vee \mathcal{C}_{\delta_0|_{betty}}\mathit{vio}_{betty}$. And it is true because the first disjunct is true. The reasoning is the same as for the item above. Note that $\langle M, w_0\rangle \vDash \neg \mathrm{K}_{betty}\mathrm{E}_{\delta_0|_{betty}}\neg \mathit{light}$, because $\langle M, w_0\rangle \vDash \neg \mathrm{K}_{betty}[\delta_0|_{betty}]\neg \mathit{light}$.

  Recall from Example 4.7 that $\langle M, w_0\rangle \vDash \mathcal{A}_{\delta_0|_{betty}}\neg \mathit{light}$, i.e., after executing action *none*, Betty is accountable for the fact that the light is off. But, we now saw that, since she did not know the state of the light, she cannot be blamed for it.

- $\langle M, w_0\rangle \vDash \neg\mathcal{B}_{\delta_0|_{alice};\delta_0|_{alice}}\neg \mathit{light}$.
  In words, it is not the case that, after executing action *none* twice, Alice is blamed for the fact that the light is off. To show it, we use the same reasoning as for the first item again, but now twice.

- $\langle M, w_0\rangle \vDash [\delta_2|_{alice}]\mathcal{B}_{\delta_0|_{betty}}\neg \mathit{light}$.
  In words, after the execution of action *tell* by Alice and then the execution of action *none*, Betty is blamed for the fact that the light is off. It is true if and only if $\langle M, w_2\rangle \vDash \mathcal{B}_{\delta_0|_{betty}}\neg \mathit{light}$. And it is true if and only if $\langle M, w_2\rangle \vDash \mathcal{C}_{\delta_0|_{betty}}\neg \mathit{light} \wedge \neg\mathcal{C}_{\delta_0|_{betty}}\neg \mathit{vio}_{betty}$. Note that the latter is true, because, in w_2 Betty knows the status of the light switch, which means that she knows that executing *none* will lead to a violation. In addition, in w_2 Betty also knows that there is another option that avoids the violation, namely, the execution of *tog*.

  In other words, once Alice tells Betty the status of the light, Betty has all she needs to avoid the violation. If she decides to execute *none* she is not only accountable for the fact that the light is off, but she knew that it would happen and she also knew how to avoid it. Therefore, she is blamed for it. ∎

4.3.3 The Relation Between Forward-Looking and Backward-Looking Responsibilities

Given the definitions for operators $\mathcal{R}$, $\mathcal{A}$ and $\mathcal{B}$ of the latter subsections, the relation between the two kinds of responsibility now seems almost evident. Indeed, it is easy to check that the following proposition is true.

Proposition 4.6. *The following axiom schemata are valid in CEDL:*

1. $(\mathcal{R}^n_G\varphi \wedge \mathrm{E}_{\delta_1|_G;\ldots;\delta_n|_G}\neg\varphi) \rightarrow \mathcal{A}_{\delta_1|_G;\ldots;\delta_n|_G}\neg\varphi$

2. $(\mathcal{R}^n_G\varphi \wedge \mathcal{C}_{\delta_1|_G;\ldots;\delta_n|_G}\neg\varphi) \rightarrow \mathcal{B}_{\delta_1|_G;\ldots;\delta_n|_G}\neg\varphi$

Proposition 4.6.1 says that: if it is obligatory for G to see to it that φ, and the sequence of actions $\delta_1;\ldots;\delta_n$ ensures $\neg\varphi$, then G is accountable for $\neg\varphi$; and Proposition 4.6.2 says that: if it is obligatory for G to see to it that φ, and the sequence of actions $\delta_1;\ldots;\delta_n$ knowingly causes $\neg\varphi$, then G is blamed for $\neg\varphi$.

4.4 The Problem of Many Hands

4.4.1 How to avoid the PMH

As said before, the problem of many hands arises in an organisation when the organisation is backward-looking responsible for some undesirable outcome but no organisation member can be held backward-looking responsible for this outcome. Let G be the set of members of a given organisation, it is formally given by:

$$\mathrm{PMH}_{\delta_G}\varphi \stackrel{\text{def}}{=} \mathcal{B}_{\delta_G}(\varphi) \wedge \bigwedge_{i\in\mathbb{A}} \neg\mathcal{B}_{\delta_i}(\varphi)$$

A formula of the form $\mathrm{PMH}_{\delta_G}\varphi$ means that 'by performing δ_G, the problem of many hands arises in group G with respect to the outcome φ'. If we follow the definitions of operator $\mathcal{B}$, we find out that we can paraphrase it as 'by performing δ_G, group G is backward-looking responsible for φ, but no agent i in G is backward-looking responsible for φ by performing δ_i'. Analysing the definitions of operator $\mathcal{B}$ a bit further we realise that the PMH may arise from three different sources: either (1) no forward-looking responsibility is ascribed to the individuals, or (2) the individuals that are forward-looking responsible for φ do not have the ability to bring it about (without the other individuals of the group), or (3) the individuals that are forward-looking responsible for φ do not have the necessary knowledge to bring about φ (without considering the knowledge of the other individuals of the group).

To address source 1, once a group is forward-looking responsible for outcome φ, we need to ascribe responsibility to at least one individual in that group. We call this individual the leader of the group. Formally we have:

(A10) $\quad \mathrm{R}^n_G\varphi \rightarrow \mathrm{R}^n_i\varphi \qquad$ (if i is the leader of G)

and where the leader i is a distinguished agent in the model (this is defined formally in the next section).

Now, to address source 2, we have to be sure that the forward-looking responsibility can be "delegated" (directly or indirectly) to the individuals that have the ability to fulfill it. This will mean that the leader should have the power to ascribe forward-looking responsibilities to other individuals of the group.

Similarly, to address source 3, we have to be sure that individuals can share information. In other words, that knowledge can be transmitted among individuals of the group.

Therefore, a group that wants to avoid the PMH should organise itself in such a way that delegation actions and information actions are possible. In management theory, some ideas of how organisations can be built are given. Here, we use the ideas of Grossi et al. (2007), and define organisations as a kind of structure.

4.4.2 Organisational structures

An organisational structure is a quadruple $\langle G, P, IR, i\rangle$ where:

- G is a group of agents in $\mathbb{A}$, representing the members of the organisation;
- P is a subset of G^2, representing a power relation among the members of the organisation;
- IR is a subset of G^2, representing an information flux relation among the members of the organisation;
- i is an agent in G, representing the leader of the group,

and that satisfies the following semantic constraints:

(SC6) $j \in P^+(i)$ (for all $j \in G$)

(SC7) $P \subseteq IR$

where P^+ is the transitive closure of P.

SC6 implies that every individual in the organisation can eventually be ascribed forward-looking responsibility for some outcome. SC7 guarantees that if an individual may be ascribed forward-looking responsibility for some outcome, this individual can also be informed about it.

Let O be an organisational structure, the models of our logic are now tuples of the form $\langle \mathrm{O}, W, R, T, I\rangle$. And its language is extended by the set of atomic formulas $\{\mathrm{Power}(i,j) \mid i,j \in G\} \cup \{\mathrm{Coord}(i,j) \mid i,j \in G\}$. The satisfaction relation is the same as before plus:

$$M, w \vDash \mathrm{Power}(i,j) \quad \text{iff} \quad (i,j) \in P$$
$$M, w \vDash \mathrm{Coord}(i,j) \quad \text{iff} \quad (i,j) \in I$$

Note that from SC7 we immediately obtain:

(A11) $\mathrm{Power}(i,j) \rightarrow \mathrm{Coord}(i,j)$

4.4.3 Organisational actions

Once organisational structures are in place, we can define the organisational actions mentioned above. The first kind of such actions consists of information actions. That is, we add the set of actions $\{\mathrm{info}(G,\varphi) : G \subseteq \mathbb{A} \text{ and } \varphi \in \mathcal{L}\}$ to the set $\mathbb{T}$ of atomic

actions of the language. For example, action $\{(i, \text{info}(G, \varphi))\}$ means 'agent i informs all members of G that φ'. We require that such actions satisfy the following two properties:

(A12) $\quad \neg[(i, \text{info}(G, \varphi))]\neg\top \rightarrow \text{K}_i\varphi$

(A13) $\quad \text{Coord}(i, j) \rightarrow [(i, \text{info}(G, \varphi))]\text{K}_j\varphi \qquad$ (for each $j \in G$)

A12 restricts the circumstances in which action $\text{info}(G, \varphi)$ is executable. We impose that agents can inform only what they know will be true after the communication act. It follows that agents cannot lie. A13 stipulates that informing actions are successful communication actions when the agents involved are appropriately related by a coordination link. This also means that the agents trust information that comes through the coordination link.

The second kind of organisational action consists of delegation actions That is, we add the set of actions:

$$\{\text{deleg}((G_1, n_1, \varphi_1), \ldots, (G_m, n_m, \varphi_m)) \mid$$
$$G_1, \ldots, G_m \subseteq \mathbb{A}, n_1, \ldots, n_m \in \mathbb{N} \text{ and } \varphi_1, \ldots, \varphi_m \in \mathcal{L}\}$$

to the set $\mathbb{T}$ of atomic actions of the language. In fact, this is a multiple-delegation action. The action $\{(i, \text{deleg}((G_1, n_1, \varphi_1), \ldots, (G_m, n_m, \varphi_m)))\}$ means 'agent i ascribes forward-looking responsibility that φ_k after n_k steps to G_k, for each $1 < k < m$'. We require that such actions satisfy:

(A14) $\quad \text{Power}(i, j) \rightarrow [(i, \text{deleg}((G_1, n_1, \varphi_1), \ldots, (G_m, n_m, \varphi_m)))]\text{K}_j\text{R}^n_k\varphi_k$
(for each $j \in G$ and each $1 < k < m$)

A14 stipulates that, when agent i delegates φ to G, it creates a forward-looking responsibility that φ for every agent in G, but only if agent i has the power to do so. Note that, in this definition of delegation, the actor of the delegation action keeps the responsibility. In addition, the agents in G also know their new responsibility. This is why power links must also be coordination links. (Recall SC7.)

4.4.4 Indirect responsibility

Note that the definition of forward-looking responsibility does not take indirect agency into account. It defines forward-looking responsibility as the case where the group must guarantee the outcome by itself. This is not well adapted for an organisation setting, where delegation actions and information actions are available. Therefore, here we redefine forward-looking responsibility below. For the sake of readability it is given in two parts. In the first part we can see that the only difference from the definition of forward-looking resposibility is the presence of operator IR. The latter operator means 'indirect responsibility'.

Definition 4.1. (Again, we show only the first three, the others are analogous.)

$$\mathrm{R}^0_G(\varphi) \stackrel{\text{def}}{=} \mathrm{O}_G(\varphi \vee \mathrm{IR}^0_G(\varphi))$$
$$\mathrm{R}^1_G(\varphi) \stackrel{\text{def}}{=} \mathrm{O}_G(\mathrm{H}_G\varphi \vee \mathrm{IR}^1_G(\varphi)) \wedge \langle\!\langle\emptyset\rangle\!\rangle \mathrm{R}^0_G(\varphi)$$
$$\mathrm{R}^2_G(\varphi) \stackrel{\text{def}}{=} \mathrm{O}_G(\mathrm{H}_G(\mathrm{H}_G\varphi \vee \mathrm{IR}^1_{G'}(\varphi)) \vee \mathrm{IR}^2_G(\varphi)) \wedge \langle\!\langle\emptyset\rangle\!\rangle \mathrm{R}^1_G(\varphi)$$

where IR^n_G is defined as:

$$\begin{aligned}
\mathrm{IR}^0_G(\varphi) \stackrel{\text{def}}{=} {} & \exists_{G'_1} \mathrm{K}_G(\mathrm{Power}(G, G'_1) \wedge \mathrm{R}^0_{G'_1}\varphi_1) \wedge \cdots \wedge \\
& \exists_{G'_m} \mathrm{K}_G(\mathrm{Power}(G, G'_m) \wedge \mathrm{R}^0_{G'_m}\varphi_m) \wedge \\
& \mathrm{K}_G((\varphi_1 \wedge \cdots \wedge \varphi_m) \to \varphi) \\
\mathrm{IR}^1_G(\varphi) \stackrel{\text{def}}{=} {} & \exists_{G'_1} \mathrm{K}_G(\mathrm{Power}(G, G'_1) \wedge \mathrm{H}_{G'_1}\varphi \wedge \mathrm{R}^1_{G'_1}\varphi_1) \wedge \cdots \wedge \\
& \exists_{G'_m} \mathrm{K}_G(\mathrm{Power}(G, G'_m) \wedge \mathrm{H}_{G'_m}\varphi \wedge \mathrm{R}^1_{G'_m}\varphi_m) \wedge \\
& \mathrm{K}_G((\varphi_1 \wedge \cdots \wedge \varphi_m) \to \varphi) \\
\mathrm{IR}^2_G(\varphi) \stackrel{\text{def}}{=} {} & \exists_{G'_1} \mathrm{K}_G(\mathrm{Power}(G, G'_1) \wedge \mathrm{H}_{G'_1}(\mathrm{H}_{G'_1}\varphi_1 \vee \mathrm{IR}^1_{G'_1}(\varphi_m)) \wedge \\
& \qquad \mathrm{R}^2_{G'_1}(\varphi_1)) \wedge \cdots \wedge \\
& \exists_{G'_m} \mathrm{K}_G(\mathrm{Power}(G, G'_m) \wedge \mathrm{H}_{G'_m}(\mathrm{H}_{G'_m}\varphi_m \vee \mathrm{IR}^1_{G'_m}(\varphi_m)) \wedge \\
& \qquad \mathrm{R}^2_{G'_m}(\varphi_m)) \wedge \\
& \mathrm{K}_G((\varphi_1 \wedge \cdots \wedge \varphi_m) \to \varphi)
\end{aligned}$$

where $\mathrm{Power}(G, G') \stackrel{\text{def}}{=} \exists_{i \in G} \forall_{j \in G'} \mathrm{Power}(i, j)$.

Informally, group G is forward-looking responsible for φ after n steps if and only if it is obligatory for G that it is able to bring about φ after n steps or that it is indirect forward-looking responsible for φ after n steps. The definition of indirect responsibility is the most complex part. Yet, it is intuitive. We consider that a group G is indirect responsible for φ if and only if there are groups $G'_1, \ldots, G'_m$ under G's power that are able to bring about $\varphi_1, \ldots, \varphi_m$, or they are indirect responsible for $\varphi_1, \ldots, \varphi_m$, and such that $\varphi_1, \ldots, \varphi_m \to \varphi$.

4.4.5 Example

Consider an organisation with two bank accounts, 1 and 2. Alice (agent *alice*) is the member of the organisation who manages the accounts. Betty (agent *betty*) is the member who normally pays the bills for the organisation. And Carol (agent c) is the director of their department, but she has no access to the accounts.

Let $\langle M, w_0 \rangle$ be a pointed CEDL model, where M is depicted in Figure 4.4. It describes a situation where Betty will pay a bill after three steps using account 1. Alice knows that, but she does not know which account Betty will use. Alice also knows that none of the accounts has enough money to pay the bill, but she does know that both

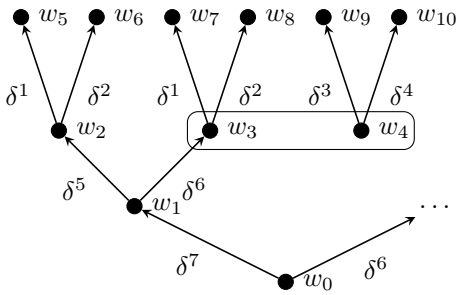


Figure 4.4 – The CEDL model $M = \langle \mathrm{O}, W, R, T, I\rangle$ for the example in Section 4.4.5: the rectangle represents $R(alice)$. The joint actions and the interpretation function I are detailed in Table 4.5. The organisation $\mathrm{O} = \langle G, P, IR, c\rangle$, where $G = \{alice, betty, c\}$, $P = \{(c, alice), (c, betty)\}$, and $IR = P \cup \{(betty, alice)\}$.

accounts together do. Group $G = \{alice, betty, c\}$ is ascribed the responsibility to have the bill paid and the balances non-negative after three steps, i.e., $\langle M, w_0\rangle \vDash \mathrm{R}^3_G(\mathrm{bal}(n) \geq 0)$.

Note that $\langle M, w_2\rangle \vDash \mathcal{C}_{\delta^2|_G}\neg(\mathrm{bal}(n) \geq 0)$. That is, at w_2, if Alice places the money on account 2, Betty pays the bill from account 1 and Carol just waits, the group G knowingly causes a negative balance. This is true because $\langle M, w_2\rangle \vDash \neg \mathrm{K}_G \mathrm{E}_{\delta^2|_G}\neg(\mathrm{bal}(n) \geq 0)$, and also $\langle M, w_2\rangle \vDash \neg\mathcal{C}_{\delta^2|_G}\neg vio_G$ i.e., it is not the case that G knows that δ^2 avoids a violation state. Therefore, $\langle M, w_2\rangle \vDash \mathcal{B}_{\delta^2|_G}\neg(\mathrm{bal}(n) \geq 0)$. A similar reasoning will show that $\langle M, w_0\rangle \vDash \mathcal{B}_{\delta^7|_G;\delta^5|_G;\delta^2|_G}\neg(\mathrm{bal}(n) \geq 0)$.

Now, suppose for a moment that $P = \emptyset$, so that action deleg does not produce its normal effect of ascribing forward-looking responsibilities to Alice and Betty. That is, under this assumption $I(vio_{alice}) = I(vio_{betty}) = \emptyset$. Then, we have $\langle M, w_2\rangle \vDash \neg\mathcal{B}_{\delta^2|_{alice}}(\mathrm{bal}(n) \geq 0)$, because there is no violation for Alice in the model. Therefore, $\langle M, w_0\rangle \vDash \neg\mathcal{B}_{\delta^7|_{alice};\delta^5|_{alice};\delta^2|_{alice}}(\mathrm{bal}(n) \geq 0)$. Similarly, there is no violation for Betty either, so $\langle M, w_0\rangle \vDash \neg\mathcal{B}_{\delta^7|_{betty};\delta^5|_{betty};\delta^2|_{betty}}(\mathrm{bal}(n) \geq 0)$. Because of Axiom A10, the violations for Carol are present in the model. However, we have $\langle M, w_2\rangle \vDash \neg\mathcal{B}_{\delta^2|_c}(\mathrm{bal}(n) \geq 0)$. The reason for this is that at w_2 Carol cannot ensure a non-violation state no matter which action she decides to execute. Therefore, $\langle M, w_0\rangle \vDash \neg\mathcal{B}_{\delta^7|_c;\delta^5|_c;\delta^2|_c}(\mathrm{bal}(n) \geq 0)$. This means that neither Alice, Betty or Carol can be held backward-looking responsible for a violation state. Thus $\langle M, w_0\rangle \vDash \mathrm{PMH}_{\delta^7|_G;\delta^5|_G;\delta^2|_G}(\neg\mathrm{bal}(n) \geq 0)$, i.e., by performing such actions, the PMH arises in group G with respect to outcome $\neg\mathrm{bal}(n) \geq 0$.

Now, let us come back to the original configuration of P. In this case, after the execution of δ^7, Alice is forward-looking responsible for $\mathrm{bal}(n) \geq 0$. Then, if δ^2 is executed at w_2, Alice is held backward-looking responsible, i.e., $\langle M, w_2\rangle \vDash \mathcal{B}^{\delta^2}_{alice}(\neg\mathrm{bal}(n) \geq 0)$. That is, the PMH does not arise in the presence of the organisational structure proposed and the organisational actions info and deleg.

Now, let us see the importance of using the new Definition 4.1 (instead of the definition in page 67) in the formalisation of this example. If we were using the old definition,

	alice	*betty*	*c*
δ^1	tr(2, 1)	pay(1)	wait
δ^2	tr(1, 2)	pay(1)	wait
δ^3	tr(2, 1)	pay(2)	wait
δ^4	tr(1, 2)	pay(2)	wait
δ^5	wait	info	wait
δ^6	wait	wait	wait
δ^7	wait	wait	deleg

	vio_G	vio_{alice}	vio_{betty}	vio_c	$\text{bal}(n) \geq 0$
w_0					•
w_1					•
w_2					•
w_3			•		•
w_4					•
w_5					•
w_6	•	•	•	•	
w_7					•
w_8	•	•	•	•	
w_9	•	•	•	•	
w_{10}					•

Table 4.5 – On the left, the joint actions for the model M in Figure 4.4. On the right, the interpretation function I for the model M in Figure 4.4. (Action info abbreviates $\text{info}(alice, [betty : \text{pay}(2)]\neg\top)$ and action deleg abbreviates $\text{deleg}((alice, 2, \text{bal}(n) \geq 0), (betty, 1, \text{H}_{alice}\text{bal}(n) \geq 0), (betty, 2, \text{paid}))$. The bullets show where the atoms are true.)

the PMH could not be avoided, because it does not take delegation and indirect agency into account. The leader of the group, Carol, has no action in her repertoire to ensure a non-violation state herself. It does not matter what she does, she will never be held backward-looking responsible. But with the new definition she can be held responsible, because δ^6 does not ensure a non-violation state for Carol, while there is an action that does, namely, δ^7, because $\langle M, w_0\rangle \vDash \text{K}_c\text{E}_{\delta_c^7}(\text{IR}^2_{\{alice,betty\}}(\text{bal}(n) \geq 0)$.

Finally, it is also interesting to notice that the information action executed by Betty on w_1 is important to make Alice aware of which account she will use to pay the bill. We invite the reader to do the exercise of considering $I = \emptyset$. In this case, action info will not produce its effect and therefore, the PMH arises.[3]

4.5 Related Work

Our formalization of forward-looking and backward-looking responsibility is based on other, more basic, "ingredients". Here, we use agents' actions, abilities, obligations and knowledge. Earlier works on the formalization of responsibility often do not deal with all of these ingredients. For instance, Santos and Carmo (1996) deal only with obligations and agents abilities. As we do here, they propose that responsibility should be paraphrased by 'obligation to ensure'. Their formalization is done by using a logic wherein one can write formulas of the form $\text{O}\mathbf{E}_i\varphi$, which stand for 'it is obligatory that i ensures φ'. The most interesting feature of this approach is the validity of the scheme $\mathbf{E}_i\mathbf{E}_j\varphi \to \mathbf{E}_i\varphi$. It expresses that 'if agent i ensures that agent j ensures φ then

[3]Note however, that under such assumption the model M must be changed. For instance, if action info does not produce any effect, worlds w_2 and w_3 should be bisimilar.

i ensures φ'. This is a useful feature for modelling indirect agency that is not present in our framework. However, Santos and Carmo's logic is not appropriate to address our problem for two reasons: it does not permit to express agents' incomplete knowledge about the situation and also does not have actions in its object language. The fact that Alice has incomplete knowledge is crucial in the example presented in Section 4.1, as well as the actions, since the problem is precisely that she needs to decide which action to execute.

With the logic for agent organisation (LAO) Dignum and Dignum (2007) Dignum and Dignum propose to formalise responsibility in terms of agents abilities. It seems to be a better alternative than Santos and Carmo's approach, because the former avoids considering that the agent is backward-looking responsible for a failure in the case the agent is not able to avoid this failure. However, as in Santos and Carmo's approach, LAO does not have actions in its object language, which means that, again, we cannot address our example using this logic either.

The formalism proposed by Grossi et al. Grossi et al. (2007) is based on a combination of dynamic and epistemic logics. Therefore, it has actions in the object language and also permits to express incomplete knowledge scenarios. However, this logic does not permit to express agents' abilities. As mentioned above, abilities are important in the definition of responsibility, and it is also important in the definition of obligations. We are not aware of other attempts to formalise the PMH in organisations.

A formalism that seems very close to CEDL is dynamic logic of agency (DLA) of Herzig and Lorini (2010a,b). CEDL and DLA appeared almost at the same time. However, their semantics, as well as their axiom systems are different. Moreover, formulas in DLA do not have the same meaning as in CEDL. For instance, a DLA formula of the form $\langle i : a\rangle\varphi$ means that 'agent i performs action a and φ is true afterwards'. This is why DLA has, for instance, axiom schema (Single): $\langle a : a\rangle\top \rightarrow [i : b]\bot$, meaning that an agent can execute only one action at a time. In fact, we have in DLA that the future is already determined: there is an actual history. This makes CEDL and DLA quite different formalisms. DLA is designed to reason about a system whose execution has already been determined, whereas CEDL is designed to reason about a system where agents are still to decide which possible execution will be undertaken.

Other approaches to responsibility, that are not based on logic, also exist. For instance, the casual models approach (Alechina et al. 2017; Chockler and Halpern 2004; Halpern 2015), describes the world in terms of variables and their values. Some variables may have a causal influence on others. This influence is modelled by a set of modifiable structural equations. Two different notions are defined: the *degree of responsibility*, which is a number in the interval $[0, 1]$ that essentially measures the minimal number of variables and changes that participate on the cause of some event; and the *degree of blame*, which is also a number in the interval $[0, 1]$ that, roughly speaking, measures the probability that those changes happen. Though it is very technical, the approach is flexible and able to model examples such as the ones we saw in this chapter. The responses to questions such as 'is the agent responsible for the outcome?' or 'is the agent blamed for the outcome?' are simply numbers on the given interval. The interpretation of those numbers are left to the user.

4.6 Conclusion

In this chapter, we saw a logic that extends PDL by epistemic formulas of the form $\mathrm{K}_i\varphi$, expressing that agent i knows that φ (similar to Grossi et al. (2007) and Herzig et al. (2000)), and also by enacted actions, i.e., formulas of the form $[\delta|_G]\varphi$, expressing that φ holds after the execution of δ by group G (similar to Herzig and Lorini (2010a,b), Royakkers (1998), and Wieringa and Meyer (1993)). It turns out that, in CEDL, formulas of the form $\langle\!\langle G \rangle\!\rangle\varphi$, expressing that G has the ability to ensure φ, can be defined as simple abbreviations. Therefore, it is possible to express operators with the same properties as the ones found in other logics of agency, such as coalition logic Pauly (2001, 2002) and ATL Alur et al. (2002), but using a simpler semantics. In addition, CEDL also enables us to give a solution to the problem of uniform strategies.

With this tool in hands, we propose a formalization of one notion of forward-looking responsibility, two notions of backward-looking responsibility, and also the relation between them. All together enables us to model the problem of many hands in organisations, and to show how organised groups of agents are more likely to avoid such problem.

We did not address decidability, complexity and expressiveness of CEDL. In particular, a deep comparison analysis between this logic and other logics of agency, such as coalition logic, ATEL, STIT Belnap et al. (2001), etc., has not been addressed here. Moreover, there is a possible extension of this logic that may be promising. The first one is the addition of temporal operators. When reasoning about responsibilities, one may want to express statements such as 'Betty must turn on the light before 7 p.m.'. This statement means that Betty may fulfill her task by turning on the light before the specified deadline. Such statement cannot be expressed in our language.

In the next chapter, we design a new logic that, as well as CEDL, can express enacted actions, agents abilities and knowledge. That new logic is different from CEDL but, on that framework, we will be able to address all the questions on the last paragraph, as well as one additional problem of CEDL. This additional problem has to do with the formalization of scenarios, such as the example on Section 4.4.5. There, we took a model that corresponds to the description of the problem and check that some formulas have their intuitive meanings. But, in reality, this kind of problem should be modeled using a set of formulas describing the initial situation and the behaviour of each action of the scenario. This kind of formalisation can be very long and error prone in CEDL. The reader is invited to check (de Lima et al. 2010b, Section 4), where the complete description of an example similar to the one on Section 4.4.5 is given. This problem has to do with the so-called *frame axioms* (Reiter 1991). We will see on the next chapter, that an elegant solution can be provided when the semantics of the logic is designed using the ideas of dynamic epistemic logic (van Ditmarsch, van der Hoek, et al. 2007).

Chapter 5

A Logic of Agent Abilities and Knowledge

As exposed on Section 4.6, several open questions remain for CEDL. In addition, there are a couple of ways in which CEDL could be improved. Because the aim of that logic includes reasoning about actions and knowledge, it seems possible to approach those matters from a dynamic epistemic logic perspective. The main advantage of this perspective is that, as shown in (van Ditmarsch, Herzig, et al. 2007), we do not need to bother with the frame problem. This permits relatively shorter scenario descriptions, thus solving one of the main issues of CEDL.

In this chapter, we design a new logic that improves CEDL. We call this new formalism alternating-time temporal dynamic epistemic logic (ATDEL). It is indeed a long name for a logic. This is motivated by the fact that ATDEL incorporates features from alternating-time temporal logic (ATL) (Alur et al. 2002) and dynamic epistemic logic (DEL) (van Ditmarsch, van der Hoek, et al. 2007). It then allows for reasoning about agents abilities and time, and also actions and knowledge, something that is not possible in CEDL and is not very common in other formalisms.

This exposition is based on the results published in (de Lima 2011, 2014). In the next section, we present some additional motivation for the design of a logic such as ATDEL. Section 5.2 presents the syntax and semantics of this new formalism. After that, Section 5.3 presents examples showing how our logic can be used to model multi-agent systems. In Section 5.4, the expressive power of ATDEL is compared to that of several other logics in the literature. In particular, it is shown that ATDEL subsumes group announcement logic (Ågotnes et al. 2010; Ågotnes and van Ditmarsch 2008), arbitrary announcement logic (Balbiani et al. 2008) and coalition announcement logic (Ågotnes et al. 2010; Ågotnes and van Ditmarsch 2008). The subsequent two sections present axiom systems and decision procedures for automated reasoning in ATDEL. We discuss computational complexity of model checking and decidability issues for the satisfiability checking problem. Section 5.7 discuss some related work and Section 5.8

concludes the chapter.

5.1 Motivation

Several formalisms aiming at modeling multi-agent systems have been proposed. The most known examples are perhaps sees-to-it-that logic (STIT) (Belnap et al. 2001; Broersen 2011), coalition logic (CL) (Pauly 2002) and alternating-time temporal logic (ATL) (Alur et al. 2002; van der Hoek and Wooldridge 2003). These formalisms allow reasoning about the abilities of the agents, i.e., about what states the agents are able to achieve. In ATL, for example, one can write the formula $\langle\!\langle G \rangle\!\rangle \varphi$, which means 'the group of agents G is able to enforce an outcome satisfying φ'. However, these logics do not enable reasoning about how the group G is able to enforce such outcomes. In other words, these logics do not enable reasoning about the actions the agents actually perform in order to enforce the outcome satisfying φ.

In the literature, we can find formalisms allowing reasoning about what outcomes agents are able to achieve and about how the agents achieve such outcomes. But frequently, they do not allow reasoning about individual actions of the agents. In other words, the actions are either exogenous or always executed jointly by all the agents of the scenario. For example, in public announcement logic (PAL) (Plaza 1989), one can write the formula $\langle \psi \rangle \mathrm{K}_i \varphi$, which means 'agent i knows that φ after the announcement of ψ'. But the announcement of φ is not "enacted" by any agent of the scenario. It is either interpreted as executed by all the agents together, or as an exogenous event. To this category, also belongs logic ES (Lakemeyer and Levesque 2005) as well as some logics of the DEL family (i.e., the dynamic epistemic logic family), such as the BMS framework (Baltag and Moss 2004), the already mentioned public announcement logic (PAL) (Plaza 1989), and public announcement logic with assignment (PALA) (van Ditmarsch et al. 2005).

Formalisms allowing reasoning about agents abilities and individual actions also exist. But their focus is on actions with epistemic effects only. To this category, belong group announcement logic (GAL) and coalition announcement logic (CAL) (Ågotnes et al. 2010; Ågotnes and van Ditmarsch 2008). Both are extensions of multi-agent epistemic logic (EL) with "enacted" public announcement operators and with group announcement operators. In GAL, one can write, e.g., the formula $\langle \mathrm{K}_i \psi \rangle \mathrm{K}_j \varphi$, which means 'agent j knows that φ after the announcement of ψ by agent i'. In addition, the formula $\langle G \rangle \varphi$ means 'there is an announcement by group G after which φ, where the other agents remain silent'. The group announcement operator in CAL is different. There, the formula $\langle\!\langle G \rangle\!\rangle \varphi$ means 'there is an announcement by group G after which φ is true, in spite of what the other agents announce'. In both formalisms though, the only kind of action present is public announcement. Such actions are a specific kind of communication actions capable of modifying the epistemic state of the agents.

Here, a new formalism called alternating-time temporal dynamic epistemic logic (ATDEL) is proposed. This logic is somewhat similar to CAL. As well as in CAL, in ATDEL, formula $\langle\!\langle G \rangle\!\rangle \mathbf{X} \varphi$ is true if and only if there is an action by group G after which φ is true, in spite of what the other agents do, which can also be read as 'the group G is

able to enforce that φ is true in the next step'. Note that all actions in CAL are public. This remains so in ATDEL. But the latter brings some improvements. First, ATDEL permits parallel actions by the same agent, i.e., in this logic, agents can also execute more than one action at the same time. Second, it contains actions with factual effects. These are actions that change the actual state of the world (and not only the epistemic state of the agents). Third, it contains temporal operators. In ATDEL, the formula $\langle\!\langle G \rangle\!\rangle \mathbf{A}\varphi$ means 'the group G is able to enforce that φ will always be true' and formula $\langle\!\langle G \rangle\!\rangle (\psi \mathbf{U} \varphi)$ means 'the group G is able to enforce that ψ is true until φ becomes true'. Fourth, the description of actions in ATDEL is concise. We argue that this permits reasonable sized multi-agent systems specifications. Fifth, a complete axiom system and decidability results for model checking and satisfiability checking are provided (those were missing for CAL). For instance, satisfiability checking is shown to be decidable when the set of available actions is finite.

5.2 The Logic

In this section, alternating-time temporal dynamic epistemic logic (ATDEL) is defined. It extends multi-agent epistemic logic (EL) with dynamic, coalition and temporal operators. In the preliminary part, EL and also the actions of ATDEL are presented. Its syntax and semantics are presented in the sequel.

5.2.1 Conflicting Actions

ATDEL extends EL, among other things, with dynamic operators taking actions as arguments. Inspired by other work on reasoning about actions (Demolombe et al. 2003; van Ditmarsch, Herzig, et al. 2007), it is assumed that every action consists of the pair formed by its executability pre-condition and the set of its post-conditions. Such way of defining actions is an alternative way to implement the successor state axioms (which are proposed, e.g., in (Lakemeyer and Levesque 2005; Reiter 1991)). It enables us to represent the actions in a simple way, which then permits system specifications with reasonable size.

An action is a pair $a = \langle \varphi, \sigma \rangle$, where $\varphi \in \mathcal{L}_{\mathrm{EL}}$ is a formula describing its executability pre-condition. This formula is sometimes also noted $\mathrm{pre}(a)$; and σ is a partial function with finite domain from $\mathbb{P}$ to $\mathcal{L}_{\mathrm{EL}}$. Each formula $\sigma(p)$, sometimes also noted $\mathrm{post}(a)(p)$, is the truth value that p will assume after the execution of a.

Example 5.1 (Two Buttons)**.** Consider a scenario with a light bulb that can be turned on and off using two different buttons. If button a is pressed, the light will turn on; and if button b is pressed, the light will turn off. Now, let the proposition *light* represent the state of the light bulb: it is true if and only if the light bulb is on. The definition of actions a and b are:

$$a = (\varphi, \{(\mathit{light} \mapsto \top)\})$$
$$b = (\psi, \{(\mathit{light} \mapsto \bot)\})$$

That is, $\mathrm{pre}(a) = \varphi$ means that the action of pressing button a is executable if and only if φ is true, whereas $\mathrm{post}(a)(p) = \top$ means that the light should turn on if the action is executed. And analogously for b. We do not bother with the contents of φ and ψ. They could, for instance, describe the state where the mechanism linking the corresponding button to the light bulb is working properly. ∎

The formalism is constructed in such a way that, if action a is executed, the truth value of p is set to true if $\mathrm{post}(a)(p)$ is true, and it is set to false if it is false. However, because we intend to use such definitions in multi-agent scenarios, we may now wonder: what should be the truth value of p if, in Example 5.1, one agent presses button a and another agent presses the button b, both at the same time? Note that the two mechanisms may very well be working properly, i.e., φ and ψ may be true at the same time.

Therefore, we must decide how to aggregate the results of several actions performed in parallel. We may take the following option. Let $a_1, \ldots, a_n$ denote the execution of a_1 to a_n in parallel, we may define:

$$\mathrm{post}(a_1, \ldots, a_n)(p) \stackrel{\mathrm{def}}{=} \bigwedge_{i=1}^{n} \mathrm{post}(a_i)(p) \vee (p \wedge \bigvee_{i=1}^{n} \mathrm{post}(a_i)(p)) \tag{5.1}$$

With this definition, the truth value of p after the execution of $a_1, \ldots, a_n$ is:

- true if every $\mathrm{post}(a_i)(p)$ is true,
- false if every $\mathrm{post}(a_i)(p)$ is false, and
- maintained if some $\mathrm{post}(a_i)(p)$ are true and some $\mathrm{post}(a_j)(p)$ are false.

Indeed, returning to our example of the light bulb, we have that, in the situation where both buttons are pressed at the same time, the light bulb will not change its state, because $\mathrm{post}(a, b)(p)$ is equivalent to p.

The solution taken above implements what is called 'shared control' in (Gerbrandy 2006). That is, the truth value of p is calculated taken all agents' actions into account. But, other solutions are equally possible. For instance, one could implement what Gerbrandy calls 'positive control'. That is, the truth value of p is set to true if at least one agent decides to do so. Or even 'negative control', i.e., p is set to false if at least one agent decides to do so.

Now, we may ask a similar question about the executability pre-condition of actions taken in parallel. But this matter seems much less controversial. We thus take the following standard approach:

$$\mathrm{pre}(a_1, \ldots, a_n) = \bigwedge_{i=1}^{n} \mathrm{pre}(a_i) \tag{5.2}$$

This means that the execution of a_1 to a_n in parallel is possible if and only if all individual actions are executable.

In what follows, we assume that these approaches for post and pre are taken. However, most of the results of this article also hold for other approaches, in particular, the

already mentioned positive and negative control. In the case of post-conditions post, the most important feature that any alternative solution should present to benefit from the results in this article is that they can be written as a formula in $\mathcal{L}_{\mathrm{EL}}$.

5.2.2 Syntax of ATDEL

The vocabulary of ATDEL contains the vocabulary of EL (i.e., a countable set $\mathbb{P}$ of propositional variables and a finite set $\mathbb{A}$ of labels denoting agents) and also a countable set E of events, which is the set of all possible combinations of executions of actions available for the agents in $\mathbb{A}$. Thus, to be able to define E, we first define what are the actions available for the agents.

Definition 5.1 (ATDEL Actions)**.** An action is a pair $a = (\varphi, \sigma)$ where $\varphi \in \mathcal{L}_{\mathrm{EL}}$ and σ is a function from a finite subset of $\mathbb{P}$ to $\mathcal{L}_{\mathrm{EL}}$. We assume, for each $i \in \mathbb{A}$, a countable set $\mathbb{T}_i$ of such actions, which is the set of actions available for agent i. It is also assumed that every set $\mathbb{T}_i$ contains the action skip $= (\top, \emptyset)$, which represents the 'no-operation' action, i.e., an action that is always executable and with no post-condition.

As said before, the first element of the pair $a = (\varphi, \sigma)$ is the executability pre-condition of a, which is also noted $\mathrm{pre}(a)$. The second element of the pair is the set of post-conditions of a, which is also noted $\mathrm{post}(a)$.

Definition 5.2 (ATDEL Events)**.** For each $i \in \mathbb{A}$ and each $a \in \mathbb{T}_i$, the pair (i, a), denotes the event of agent i executing action a. The set E of all possible events is the set of all subsets of the set $\{(i, a) : i \in \mathbb{A} \text{ and } a \in \mathbb{T}_i\}$.

Note that this definition allows events where the same agent executes more than one action at the same time. Also note that $e_{\mathbb{A}} = e$ and $e_{\emptyset} = \emptyset$ are also valid events of E. Moreover, let $G \subseteq \mathbb{A}$ be a group of agents, and let $e \in E$ be a possible event, we use e_G to denote the G's part of e, i.e., $e_G = \{(i, a) : (i, a) \in e \text{ and } i \in G\}$. And finally, we sometimes use skip_G to denote the event $\{(i, \mathrm{skip}) : i \in G\} \in E$.

Definition 5.3 (ATDEL Language)**.** The language $\mathcal{L}_{\mathrm{ATDEL}}$ of ATDEL is the set of formulas φ defined by the following BNF:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{K}_i\varphi \mid [e_G]\varphi \mid \langle\!\langle G \rangle\!\rangle \mathbf{X}\varphi \mid \langle\!\langle G \rangle\!\rangle \mathbf{A}\varphi \mid \langle\!\langle G \rangle\!\rangle (\varphi \mathbf{U} \varphi)$$

where p ranges over $\mathbb{P}$, i ranges over $\mathbb{A}$, G ranges over $2^{\mathbb{A}}$, and e ranges over E.

The fragment of $\mathcal{L}$ without operators $\langle\!\langle\ \rangle\!\rangle \mathbf{U}$ and $\langle\!\langle\ \rangle\!\rangle \mathbf{A}$ is called the 'next-fragment of ATDEL and is noted $\mathcal{L}_{\mathrm{X}}$.

In what follows, the common abbreviations for $\vee$, $\rightarrow$, $\leftrightarrow$ and $\bot$ are used. We also use the abbreviations for the duals of $[\ ]$ and $\langle\!\langle\ \rangle\!\rangle \mathbf{X}$. They are defined by $\langle e_G \rangle \varphi \stackrel{\mathrm{def}}{=} \neg[e_G]\neg\varphi$ and $[\![G]\!]\mathbf{X}\varphi \stackrel{\mathrm{def}}{=} \neg\langle\!\langle G \rangle\!\rangle \mathbf{X}\neg\varphi$, respectively. In addition, we sometimes use $\langle\!\langle G \rangle\!\rangle \mathbf{X}^n$, for $n \geq 0$, wich stands for a sequence of n operators $\langle\!\langle G \rangle\!\rangle \mathbf{X}$, i.e., $\langle\!\langle G \rangle\!\rangle \mathbf{X}^0 \varphi \stackrel{\mathrm{def}}{=} \varphi$ and $\langle\!\langle G \rangle\!\rangle \mathbf{X}^{n+1} \varphi \stackrel{\mathrm{def}}{=} \langle\!\langle G \rangle\!\rangle \mathbf{X} \langle\!\langle G \rangle\!\rangle \mathbf{X}^n \varphi$.

The intended meaning of $e_G = \{(i_1, a_1), \ldots, (i_n, a_n)\}$ is 'all the agents in $\{i_1, \ldots, i_n\}$ execute their corresponding actions in $\{a_1, \ldots, a_n\}$ simultaneously'. As well as in other strategic logics, it is assumed that actions are executed synchronously and that the occurrence of each event $e_{\mathbb{A}}$ corresponds to a "time step". But also note that the same agent may appear more than once in $\{i_1, \ldots, i_n\}$. This means that each agent may execute more than one action at the same time. For example, it enables modeling scenarios where the same agent can press two different buttons, send two different messages, etc., simultaneously.

The intended meaning of formula $[e_G]\varphi$ is 'after every possible occurrence of e_G, φ is true'. The intended meaning of formula $\langle\!\langle G \rangle\!\rangle \mathbf{X}\varphi$ is 'group G is able to enforce that φ is true in the next step'. Thus, the intended meaning of its dual $[\![G]\!]\mathbf{X}\varphi$ is 'it is not the case that group G is able to enforce that $\neg\varphi$ is true in the next step', or, equivalently, 'group G is not able to avoid that φ is true in the next step'. Moreover, the intended meaning of formulas of the form $\langle\!\langle G \rangle\!\rangle \mathbf{A}\varphi$ is 'group G is able to enforce that φ will always be true', while formulas of the form $\langle\!\langle G \rangle\!\rangle (\psi \mathbf{U} \varphi)$ is intended to mean 'group G is able to enforce that ψ is true until φ becomes true'.

We present some admissible inference rules for ATDEL in Section 5.5.1. Some of these rules are formulated using 'necessity forms'. These are defined as the set of forms η defined by the following BNF:

$$\eta ::= \sharp \mid \varphi \to \eta \mid \mathrm{K}_i \eta \mid [e_G]\eta$$

where $\sharp$ is a special symbol (that appears only once in a necessity form), φ ranges over $\mathcal{L}_{\mathrm{ATDEL}}$, i ranges over $\mathbb{A}$, G ranges over $2^{\mathbb{A}}$ and e ranges over E.

If η is a necessity form, then $\eta(\varphi)$ is obtained from η by substituting φ for $\sharp$ in η. For example, $\sharp$ is a necessity form (the most basic one). Thus, so is $\mathrm{K}_i\sharp$ and also $p \to \mathrm{K}_i\sharp$, as well as $[e_{\mathbb{A}}](p \to \mathrm{K}_i\sharp)$. Therefore $[e_{\mathbb{A}}](p \to \mathrm{K}_i\sharp)(p \wedge q)$ amounts to $[e_{\mathbb{A}}](p \to \mathrm{K}_i(p \wedge q))$.

The role of such forms in the axiom system of ATDEL is explained in Section 5.5.1.

5.2.3 Semantics of ATDEL

Formulas in $\mathcal{L}_{\mathrm{ATDEL}}$ are interpreted using epistemic models (i.e., Kripke models respecting axioms (T) and (5)). The dynamic operators of ATDEL are interpreted using 'model updates'. The update of M by event e modifies M in two ways: the possible worlds not satisfying its occurrence pre-condition are removed and the truth value of propositional variables are changed according to its post-conditions. Formally, it is defined as follows.

Definition 5.4 (ATDEL Update)**.** The update of the epistemic model $M = \langle W, R, V \rangle$ by the event $e \in E$ is the new model $M|e = \langle W|e, R|e, V|e \rangle$ where:

$$\begin{aligned} W|e &= \{w \mid M, w \vDash \mathrm{pre}(e)\} \\ R|e(i) &= R(i) \cap (W|e \times W|e) \\ V|e(p) &= \{w \mid \langle M, w \rangle \vDash \mathrm{post}(e)(p)\} \cap W|e \end{aligned}$$

and where $\mathrm{pre}(e)$ aggregates the pre-conditions of all actions in e, as defined Section 5.2.1, and $\mathrm{post}(e)(p)$ aggregates the post-conditions of the individual actions in e, also as defined in Section 5.2.1.

In the sequel, the satisfaction relation of ATDEL is defined.

Definition 5.5 (ATDEL Satisfaction Relation)**.** The satisfaction relation $\vDash$ between pointed epistemic models $\langle M, w\rangle$ and formulas in $\mathcal{L}_{\text{ATDEL}}$ is defined as usual for formulas in $\mathcal{L}_{\text{EL}}$ plus:

$$
\begin{array}{lll}
\langle M, w\rangle \vDash [e_G]\varphi & \text{iff} & \text{for all } e' \in E, \\
 & & \text{if } \langle M, w\rangle \vDash \text{pre}(e_G \cup e'_{\overline{G}}) \text{ then } \langle M|(e_G \cup e'_{\overline{G}}), w\rangle \vDash \varphi \\
\langle M, w\rangle \vDash \langle\!\langle G\rangle\!\rangle \mathbf{X}\varphi & \text{iff} & \text{there is } e \in E \\
 & & \text{such that } \langle M, w\rangle \vDash \neg[e_G]\bot \text{ and } \langle M, w\rangle \vDash [e_G]\varphi \\
\langle M, w\rangle \vDash \langle\!\langle G\rangle\!\rangle \mathbf{A}\varphi & \text{iff} & \text{for all } n \in \mathbb{N}, \text{ if } n \geq 0 \text{ then } \langle M, w\rangle \vDash \langle\!\langle G\rangle\!\rangle \mathbf{X}^n\varphi \\
\langle M, w\rangle \vDash \langle\!\langle G\rangle\!\rangle (\psi \mathbf{U} \varphi) & \text{iff} & \text{there is } n \in \mathbb{N} \text{ s.t. } n \geq 0 \text{ and } \langle M, w\rangle \vDash \langle\!\langle G\rangle\!\rangle \mathbf{X}^n\varphi \text{ and} \\
 & & \text{for all } m \in \mathbb{N}, \text{ if } 0 \leq m < n \text{ then } \langle M, w\rangle \vDash \langle\!\langle G\rangle\!\rangle \mathbf{X}^m(\neg\varphi \wedge \psi)
\end{array}
$$

Sometimes, to avoid confusion, we also note this satisfaction relation $\vDash_{\text{ATDEL}}$.

The interpretation of the operator [ ] embeds a quantification over events. A formula of the form $[e_G]\varphi$ is true if and only if the occurrence of G's part of e leads, necessarily, to an outcome satisfying φ. In other words, it is true if and only if the occurrence of e_G with any possible "completion" of it $e'_{\overline{G}}$ leads to an outcome satisfying φ.

Remark 5.1. Thus, following the definition given above, we have:

$$
\begin{array}{lll}
\langle M, w\rangle \vDash [e_{\mathbb{A}}]\varphi & \text{iff} & \text{for all } e' \in E, \\
 & & \text{if } \langle M, w\rangle \vDash \text{pre}(e_{\mathbb{A}} \cup e'_{\emptyset}) \text{ then } \langle M|(e_{\mathbb{A}} \cup e'_{\emptyset}), w\rangle \vDash \varphi \\
 & \text{iff} & \text{if } \langle M, w\rangle \vDash \text{pre}(e_{\mathbb{A}}) \text{ then } \langle M|e_{\mathbb{A}}, w\rangle \vDash \varphi
\end{array}
$$

because $e'_{\emptyset} = \emptyset$ for all $e' \in E$.

Remark 5.2. And we also have:

$$
\begin{array}{lll}
\langle M, w\rangle \vDash [e_{\emptyset}]\varphi & \text{iff} & \text{for all } e' \in E, \\
 & & \text{if } \langle M, w\rangle \vDash \text{pre}(e_{\emptyset} \cup e'_{\mathbb{A}}) \text{ then } \langle M|(e_{\emptyset} \cup e'_{\mathbb{A}}), w\rangle \vDash \varphi \\
 & \text{iff} & \text{for all } e' \in E, \text{ if } \langle M, w\rangle \vDash \text{pre}(e'_{\mathbb{A}}) \text{ then } \langle M|e'_{\mathbb{A}}, w\rangle \vDash \varphi
\end{array}
$$

again, because $e_{\emptyset} = \emptyset$ for all $e \in E$.

Remark 5.3. Let us also explain what happens if the executability pre-condition of an action is false. Assume an event $e = \{(i_1, a_1), \ldots, (i_n, a_n)\}$, where $\langle M, w\rangle \nvDash \text{pre}(a_k)$, for some a_k, $1 \leq k \leq n$. Using (5.2), we have $\langle M, w\rangle \nvDash \text{pre}(e)$. Using the definitions just given, we have: $\langle M, w\rangle \vDash [e]\bot$. The latter formula reads 'after every possible execution of e, $\bot$ is true'. This is as it is supposed to be, since we have just assumed that there is no possible execution of e.

The interpretation of the operator $\langle\!\langle\ \rangle\!\rangle \mathbf{X}$ is based on the interpretation of [ ]. It stipulates that the formula $\langle\!\langle G\rangle\!\rangle \mathbf{X}\varphi$ is true if and only if there is an event $e \in E$ such that G's part of e can occur and such that its occurrence, necessarily, leads to an outcome satisfying φ.[1]

[1] Note that this definition is slightly different from that of (de Lima 2011). The definition there stipulates that such formula is true if and only if there is an event $e \in E$ such that G's part of e necessarily leads to an outcome satisfying φ, but its occurrence does not need to be possible. This leads to slightly different axioms for the operator $\langle\!\langle\ \rangle\!\rangle \mathbf{X}$.

As usual, a formula $\varphi \in \mathcal{L}_{\text{ATDEL}}$ is valid in ATDEL, noted $\vDash \varphi$, if and only if every pointed epistemic model $\langle M, w\rangle$ satisfies φ. A formula $\varphi \in \mathcal{L}_{\text{ATDEL}}$ is satisfiable in ATDEL, if and only if there is a pointed epistemic model $\langle M, w\rangle$ that satisfies φ, i.e., $\nvDash \neg\varphi$.

5.3 Examples

We show in this section two examples of scenarios which can be modeled using ATDEL.

Example 5.2 (Light bulb and light switch)**.** ATDEL can be used to reason about collaborative agency. To see it, we consider again the scenario of Example 4.4. To formalize this in ATDEL, let the set of agents be $\mathbb{A} = \{alice, betty\}$ and the actions available for them be:

$$\begin{aligned}
\mathbb{T}_{alice} &= \{\text{skip}, tellon, telloff\}\\
\mathbb{T}_{betty} &= \{\text{skip}, tog\}\\
&\text{where:}\\
tog &= (\top, \{(light \mapsto \neg light)\})\\
tellon &= (\text{K}_{alice} light, \emptyset)\\
telloff &= (\text{K}_{betty} \neg light, \emptyset)
\end{aligned}$$

Action *tog* is available only for Betty while actions *tellon* and *telloff* are available only for Alice. Note that the latter two actions work as the public announcements that the light bulb is on and off, respectively. We do not have, as in Chapter 4, the action *tell* here. This is so because this logic does not contain non-deterministic actions. Note that *tell* can be seen as the non-deterministic choice between *tellon* and *telloff*.

The set of possible events E is formed by all combinations of these actions. For the considerations below, we will use only the following ones:

$$\begin{aligned}
e &= \{(alice, tellon), (betty, \text{skip})\}\\
e' &= \{(alice, telloff), (betty, \text{skip})\}\\
f &= \{(alice, \text{skip}), (betty, tog)\}\\
f' &= \{(alice, \text{skip}), (betty, \text{skip})\}
\end{aligned}$$

A model for this example is depicted in Figure 5.1. In the actual world w_0, the light is off ($\neg light$), Alice knows it, but Betty does not. After the occurrence of e', the model is updated: the world where $\text{pre}(e')$ is false is removed. In the resulting model, both Alice and Betty know that the light is off. Then, after the occurrence of f, the truth value of *light* is changed. In the resulting model, both Alice and Betty know that the light is now on.

It is easy to check that every pointed model $\langle M, w\rangle$ satisfies:

$$\neg light \to [e'_{\{alice, betty\}}][f_{\{alice, betty\}}](\text{K}_{alice} light \wedge \text{K}_{betty} light)$$

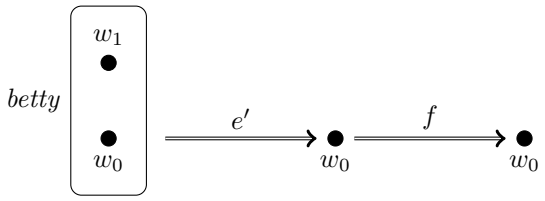


Figure 5.1 – Example of model for the light bulb and light switch scenario

which means that 'if the light is off, then both agents will know that the light will be on after Alice telling that it is off and Betty toggling the switch'. Therefore, we have that every pointed model also satisfies:

$$\neg \mathit{light} \to \langle\!\langle \{\mathit{alice}, \mathit{betty}\} \rangle\!\rangle \mathbf{X} \langle\!\langle \{\mathit{alice}, \mathit{betty}\} \rangle\!\rangle \mathbf{X} (\mathrm{K}_{\mathit{alice}} \mathit{light} \wedge \mathrm{K}_{\mathit{betty}} \mathit{light})$$

which means that 'if the light is off, then after two steps Alice and Betty are able to enforce an outcome where both of them know that the light is on'. Analogously, it is easy to check that every pointed model also satisfies:

$$\mathit{light} \to [e_{\{\mathit{alice},\mathit{betty}\}}][f'_{\{\mathit{alice},\mathit{betty}\}}](\mathrm{K}_{\mathit{alice}} \mathit{light} \wedge \mathrm{K}_{\mathit{betty}} \mathit{light})$$

which implies that we also have:

$$\mathit{light} \to \langle\!\langle \{\mathit{alice}, \mathit{betty}\} \rangle\!\rangle \mathbf{X} \langle\!\langle \{\mathit{alice}, \mathit{betty}\} \rangle\!\rangle \mathbf{X} (\mathrm{K}_{\mathit{alice}} \mathit{light} \wedge \mathrm{K}_{\mathit{betty}} \mathit{light})$$

All together, this implies that every pointed model satisfies:

$$\langle\!\langle \{\mathit{alice}, \mathit{betty}\} \rangle\!\rangle \mathbf{X} \langle\!\langle \{\mathit{alice}, \mathit{betty}\} \rangle\!\rangle \mathbf{X} (\mathrm{K}_{\mathit{alice}} \mathit{light} \wedge \mathrm{K}_{\mathit{betty}} \mathit{light})$$

And finally, it implies that every pointed model satisfies

$$\langle\!\langle \{\mathit{alice}, \mathit{betty}\} \rangle\!\rangle (\top \mathbf{U} (\mathrm{K}_{\mathit{alice}} \mathit{light} \wedge \mathrm{K}_{\mathit{betty}} \mathit{light}))$$

which in words means that whatever the initial situation is, Alice and Betty are able to enforce that, eventually, both know that the light is on. ∎

It is perhaps worthwhile to stress that the descriptions in terms of functions pre and post above are all that is needed to formalize the actions of this scenario. This is one of the advantages of using such functions to describe actions. In other formalisms, a relatively large number of formulas would be needed to achieve the same. This is, for instance, the case for the formalism presented on Chapter 4. Also note that, if one wants to add more actions to the scenario, including actions with different epistemic effects or even actions that change the truth value of other additional propositional variables, the description of *tellon*, *telloff* and *tog* do not need to be changed.

Example 5.3 (Tic-tac-toe)**.** To illustrate that ATDEL can also be used to model competitive agency, such as in game-like scenarios, we consider a formalization of the game

tic-tac-toe. We use capital letters to name each cell in the board from the left to the right and from the top to the bottom (i.e., A names the leftmost top cell, B names the middle top cell, ..., and I names the rightmost bottom cell). Then, we assume some propositional variables describing the situation of the game, e.g., p_{CA} means 'there is a cross in cell A', and two propositional variables expressing which player has the right to play: q_C means 'it is Cross' turn to play' and q_O means 'it is Nought's turn to play'. Finally, we assume some actions describing the possible plays, e.g., a_{OB} means 'plays a nought in cell B'. These actions can be defined as follows. For all $x \in \{C, O\}$ and $y \in \{A, \ldots, I\}$:

$$\begin{aligned} \mathrm{pre}(a_{xy}) &= q_x \wedge \neg p_{Cy} \wedge \neg p_{Oy} \\ \mathrm{post}(a_{xy}) &= \{(p_{xy} \mapsto \top), (q_{\overline{x}} \mapsto \top), (q_x \mapsto \bot)\} \end{aligned}$$

where $\overline{x}$ means 'the opposite player', i.e., $\overline{C} = O$ and $\overline{O} = C$. For instance, it follows from these definitions that player x can play x in cell y if and only if it is x's turn and there is no cross nor nought in the cell y. The actions a_{Cy} are only available for player C, while actions a_{Oy} are only available for player O. Finally, let some possible events be:

$$\begin{aligned} e_{\{C,O\}} &= \{(C, a_{CI}), (O, \mathrm{skip})\} \\ e'_{\{C,O\}} &= \{(C, a_{CD}), (O, \mathrm{skip})\} \\ f_{\{C,O\}} &= \{(C, \mathrm{skip}), (O, a_{OI})\} \end{aligned}$$

Now, let us suppose an already started match which looks like the following picture, where it is Cross' turn to play:

X		O
	X	O
O		

Assume a pointed model $\langle M, w\rangle$ satisfying this situation, i.e., assume:

$$\begin{aligned} \langle M, w\rangle \vDash\ & q_C \wedge p_{CA} \wedge \neg p_{CB} \wedge \neg p_{CC} \wedge \\ & \neg p_{CD} \wedge p_{CE} \wedge \neg p_{CF} \wedge \\ & \neg p_{CG} \wedge \neg p_{CH} \wedge \neg p_{CI} \wedge \\ & \neg p_{OA} \wedge \neg p_{OB} \wedge p_{OC} \wedge \\ & \neg p_{OD} \wedge \neg p_{OE} \wedge p_{OF} \wedge \\ & p_{OG} \wedge \neg p_{OH} \wedge \neg p_{OI} \end{aligned}$$

It is easy to check that $\langle M, w\rangle$ also satisfies $[e_{\{C\}}]p_{CI}$, which means that p_{XI} becomes true after such action and, therefore, player C wins. Note that such formula implies $\langle\!\langle\{C\}\rangle\!\rangle \mathbf{X} p_{CI}$, which means that player C can win in one step. But we also have that this model satisfies $[e'_{\{C\}}][f_{\{O\}}]p_{OI}$, which means that after some other play by player C, player O can win. Note that this implies $\langle\!\langle C\rangle\!\rangle \mathbf{X} \langle\!\langle O\rangle\!\rangle \mathbf{X} p_{OI}$, which means that X can put O in a position where O can win the game. ∎

5.4 Expressiveness

In this section, we compare the expressive power of ATDEL to that of several different members of the DEL family. This is done by presenting different versions of ATDEL that are at least as expressive as the various members of the DEL family. We consider five different logics in that family: public announcement logic (PAL), public announcement logic with assignment (PALA), group announcement logic (GAL), coalition announcement logic (CAL), and arbitrary announcement logic (APAL). First though, we need to define expressiveness, which is in turn defined in terms of model distinguishablility.

Definition 5.6 (Model Distinguishability)**.** Logic L distinguishes two pointed epistemic models $\langle M, w\rangle$ and $\langle M', w'\rangle$, noted $\langle M, w\rangle \not\equiv_{\mathrm{L}} (M', w')$, if and only if there is a formula φ in the language of L such that $\langle M, w\rangle \models_{\mathrm{L}} \varphi$ and $\langle M', w'\rangle \not\models_{\mathrm{L}} \varphi$.

Definition 5.7 (Expressiveness)**.** Let L_1 and L_2 be two logics interpreted over the class of pointed epistemic models. Then, L_2 is at least as expressive as L_1, noted $\mathrm{L}_1 \leq \mathrm{L}_2$, if and only if L_2 distinguishes at least the same models that L_1 distinguishes, i.e., $\mathrm{L}_1 \leq \mathrm{L}_2$, if and only if, for all pairs of pointed epistemic models $\langle M, w\rangle$ and $\langle M', w'\rangle$ if $\langle M, w\rangle \not\equiv_{\mathrm{L}_1} \langle M', w'\rangle$ then $\langle M, w\rangle \not\equiv_{\mathrm{L}_2} \langle M', w'\rangle$. We also use $\mathrm{L}_1 = \mathrm{L}_2$ to express that both $\mathrm{L}_1 \leq \mathrm{L}_2$ and $\mathrm{L}_2 \leq \mathrm{L}_1$ are true.

5.4.1 ATDEL vs. PAL and PALA

Clearly, ATDEL is at least as expressive as EL. Moreover, PALA is a conservative extension of PAL, thus, PAL $\leq$ PALA. In addition, it is a known fact that every formula in $\mathcal{L}_{\mathrm{PALA}}$ (thus also in PAL) has an equivalent formula in $\mathcal{L}_{\mathrm{EL}}$ (Kooi 2007, Theorem 15). Therefore, we immediately have the following result.

Proposition 5.4. PAL = PALA = EL $\leq$ ATDEL

5.4.2 ATDEL vs. APAL

The language $\mathcal{L}_{\mathrm{APAL}}$ of APAL contains $\mathcal{L}_{\mathrm{EL}}$ plus formulas of the form $[\psi]\varphi$, which are read 'after the public announcement of ψ, φ is true'; and also formulas of the form $\Box\varphi$, which are read 'after every public announcement, φ is true'. Their interpretation is given by:

$$\begin{array}{lll} M, w \models_{\mathrm{APAL}} [\psi]\varphi & \text{iff} & M, w \models_{\mathrm{APAL}} \psi \text{ implies } M|\psi, w \models_{\mathrm{APAL}} \varphi \\ M, w \models_{\mathrm{APAL}} \Box\varphi & \text{iff} & \text{for all } \psi \in \mathcal{L}_{\mathrm{EL}}, M, w \models_{\mathrm{APAL}} [\psi]\varphi \end{array}$$

where $M|\psi = \langle W|\psi, R|\psi, V|\psi\rangle$ is the update of M by the public announcement of ψ, defined as follows.

$$\begin{aligned} W|\psi &= \{w \mid M, w \models_{\mathrm{APAL}} \psi\} \\ R|\psi(i) &= R(i) \cap (W|\psi \times W|\psi) \\ V|\psi(p) &= \{w \mid M, w \models_{\mathrm{APAL}} p\} \cap W|\psi \end{aligned}$$

Note that formulas of the form $[\Box\psi]\varphi$ are permitted in APAL. In the sequel, however, we do not deal with such formulas. That is, we consider a fragment of $\mathcal{L}_{\text{APAL}}$ where public announcements are formulas in $\mathcal{L}_{\text{EL}}$. In other words, let $\mathbb{P}$ be a countable set of propositional variables and let $\mathbb{A}$ be a finite set of labels denoting agents, the language $\mathcal{L}_{\text{APALR}}$ considered here is the set of formulas φ defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{K}_i\varphi \mid [\psi]\varphi \mid \Box\varphi$$
$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \mathrm{K}_i\psi$$

where p ranges over $\mathbb{P}$ and i ranges over $\mathbb{A}$.

We call the corresponding logic APALR. This is not an uninteresting fragment. Note that formulas of the form $[\psi]\varphi$, where ψ itself contains public announcements, can be reduced to equivalent formulas in $\mathcal{L}_{\text{EL}}$ (cf. (Kooi 2007)). In addition, the semantic interpretation of formulas of the form $\Box\psi$ uses a quantification over $\mathcal{L}_{\text{EL}}$ in order to avoid a circular definition (cf. (Balbiani et al. 2008)). Therefore, even though formulas of the form $[\Box\psi]\varphi$ are allowed in the full language of APAL, the public announcement in it (i.e., formula $\Box\psi$) does not count as a possible announcement for the interpretation of $\Box\varphi$. Here, we avoid to deal with such announcements just because the action pre-conditions and post-conditions in ATEL are restricted to formulas in $\mathcal{L}_{\text{EL}}$. We leave as an open question weather such announcements can be expressed in ATDEL.

Now, let us compare the expressiveness of APALR to that of ATDEL_1, which is the version of ATDEL that respects the following condition.

(i) For all $i \in \mathbb{A}$, $\mathbb{T}_i = \{(\varphi, \emptyset) \mid \varphi \in \mathcal{L}_{\text{EL}}\}$.

The key idea is to make ATDEL_1 respect two restrictions of APALR, namely, (1) public announcements are the only kind of action allowed (which is already the case in APAL) and (2) only formulas in $\mathcal{L}_{\text{EL}}$ can be announced.

The translation mltr from $\mathcal{L}_{\text{APALR}}$ to the language of ATDEL_1 is defined as follows:

$$\begin{aligned}
&\mathrm{mltr}(\varphi) \stackrel{\text{def}}{=} \varphi && (\text{if } \varphi \in \mathcal{L}_{\text{EL}})\\
&\mathrm{mltr}([\varphi_1]\varphi_2) \stackrel{\text{def}}{=} [\{(i, (\varphi_1, \emptyset))\} \cup \mathrm{skip}_{\mathbb{A}}]\,\mathrm{mltr}(\varphi_2)\\
&\mathrm{mltr}(\Box\varphi) \stackrel{\text{def}}{=} [\emptyset]\,\mathrm{mltr}(\varphi)
\end{aligned}$$

where the agent i in the second clause is arbitrarily chosen among $\mathbb{A}$.

The intuition behind the second clause is that public announcements φ_1 are translated to actions $(\varphi_1, \emptyset)$, enacted by one of the agents, while all the other agents remain silent.

We obtain the following expressiveness result.[2]

Proposition 5.5. $\text{APALR} \leq \text{ATDEL}_1$.

[2]For readability, several proofs are omitted in this chapter. They can be found in (de Lima 2014).

5.4.3 ATDEL vs. GAL

The language $\mathcal{L}_{\mathrm{GAL}}$ of GAL contains $\mathcal{L}_{\mathrm{EL}}$ plus formulas of the form $[\psi]\varphi$, which are both read and interpreted as in APAL; and also formulas of the form $\langle G\rangle\varphi$, which are read 'there is an announcement by group G after which φ, where the other agents remain silent'. The satisfaction relation $\vDash_{\mathrm{GAL}}$ is the usual one plus:

$$M, w \vDash_{\mathrm{GAL}} \langle G\rangle\varphi \qquad \text{iff} \qquad \text{there exists a set } \{\psi_i \mid i \in G\} \subseteq \mathcal{L}_{\mathrm{EL}} \text{ s.t.}$$
$$M, w \vDash_{\mathrm{GAL}} \bigwedge_{i\in G} \mathrm{K}_i\psi \text{ and } M|\bigwedge_{i\in G} \mathrm{K}_i\psi_i, w \vDash_{\mathrm{GAL}} \varphi$$

where $M|\psi$ is defined as in APAL.

Note that formulas of the form $[\langle G\rangle\psi]\varphi$ are permitted in GAL. For analogous reasons to the ones in Section 5.4.2, we consider a fragment of $\mathcal{L}_{\mathrm{GAL}}$ where public announcements are formulas in $\mathcal{L}_{\mathrm{EL}}$. In other words, let $\mathbb{P}$ be a countable set of propositional variables and let $\mathbb{A}$ be a finite set of labels denoting agents, the language $\mathcal{L}_{\mathrm{GALR}}$ considered here is the set of formulas φ defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{K}_i\varphi \mid [\psi]\varphi \mid \langle G\rangle\varphi$$
$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \mathrm{K}_i\psi$$

where p ranges over $\mathbb{P}$, i ranges over $\mathbb{A}$ and G ranges over $2^{\mathbb{A}}$.

We call the corresponding logic GALR. Let us compare its expressiveness to that of ATDEL_2, which is the version of ATDEL with the finite set $\mathbb{A} \cup \{s\}$ of labels denoting agents and which respects following condition.

(i) The set of available actions for s is $\mathbb{T}_s = \{(\varphi, \emptyset) \mid \varphi \in \mathcal{L}_{\mathrm{EL}}\}$ and, for each agent $i \in \mathbb{A}$, the set of available actions for i is $\mathbb{T}_i = \{(\mathrm{K}_i\varphi, \emptyset) \mid \mathrm{K}_i\varphi \in \mathcal{L}_{\mathrm{EL}}\}$.

The key idea is to have a 'special agent' s that is the only one acting whenever a public announcement is made. In addition, the condition above also makes ATDEL_2 respect three restrictions of GALR (which are already the case in GAL), namely, (1) public announcements are the only kind of action allowed, (2) each agent in $\mathbb{A}$ can only announce what is known and (3) only formulas in $\mathcal{L}_{\mathrm{EL}}$ can be announced by the agents in $\mathbb{A}$.

The translation function mltr from $\mathcal{L}_{\mathrm{GALR}}$ to the language of ATDEL_2 is defined as follows.

$$\mathrm{mltr}(\varphi) \stackrel{\mathrm{def}}{=} \varphi \qquad (\text{if } \varphi \in \mathcal{L}_{\mathrm{EL}})$$
$$\mathrm{mltr}([\varphi_1]\varphi_2) \stackrel{\mathrm{def}}{=} [\{(s, (\mathrm{mltr}(\varphi_1), \emptyset))\} \cup \mathrm{skip}_{\mathbb{A}}]\,\mathrm{mltr}(\varphi_2)$$
$$\mathrm{mltr}(\langle G\rangle\varphi) \stackrel{\mathrm{def}}{=} \langle \mathrm{skip}_{\overline{G\cup\{s\}}}\rangle\,\mathrm{mltr}(\varphi)$$

The intuition behind the second clause is that public announcements φ_1 are translated to actions $(\varphi_1, \emptyset)$ enacted by the agent s, while all the other agents remain silent. We must have a special agent s here because the agents in $\mathbb{A}$ are not allowed to announce what they do not know. To be able to express, for instance, the formula $[p]\varphi$

in ATDEL_2, we need to add an agent capable to announce p, which is not necessarily known.

We obtain the following result.

Proposition 5.6. $\mathrm{GALR} \leq \mathrm{ATDEL}_2$.

5.4.4 ATDEL vs. CAL

The language $\mathcal{L}_{\mathrm{CAL}}$ of CAL contains $\mathcal{L}_{\mathrm{EL}}$ plus formulas of the form $[\psi]\varphi$, which are both read and interpreted as in APAL; and also formulas of the form $\langle\!\langle G \rangle\!\rangle \varphi$, which read 'there is an announcement by group G after which φ, in spite of what the other agents announce'. Its interpretation is given by:

$$
\begin{array}{lll}
M, w \vDash_{\mathrm{CAL}} \langle\!\langle G \rangle\!\rangle \varphi & \text{iff} & \text{for every } i \in G \text{ there exists } \psi_i \in \mathcal{L}_{\mathrm{EL}} \\
 & & \text{s.t. for every } \psi_j \in \mathcal{L}_{\mathrm{EL}} \text{ for each } j \in \overline{G}, \\
 & & M, w \vDash_{\mathrm{CAL}} \bigwedge\limits_{i \in G} \mathrm{K}_i \psi_i \wedge [\bigwedge\limits_{k \in \mathbb{A}} \mathrm{K}_k \psi_k] \varphi
\end{array}
$$

Again, note that formulas of the form $[\langle\!\langle G \rangle\!\rangle \psi]\varphi$ are permitted in CAL. For analogous reasons to the ones in Section 5.4.2, we consider a fragment of $\mathcal{L}_{\mathrm{CAL}}$ where public announcements are conjunctions of formulas in $\mathcal{L}_{\mathrm{EL}}$ which are known by some agent. In other words, let $\mathbb{P}$ be a countable set of propositional variables and let $\mathbb{A}$ be a finite set of labels denoting agents, the language $\mathcal{L}_{\mathrm{CALR}}$ considered here is the set of formulas φ defined by the following BNF:

$$
\begin{array}{l}
\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{K}_i \varphi \mid [\psi]\varphi \mid \langle\!\langle G \rangle\!\rangle \varphi \\
\psi ::= \mathrm{K}_i \chi \mid \psi \wedge \psi \\
\chi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \mathrm{K}_i \psi
\end{array}
$$

where p ranges over $\mathbb{P}$, i ranges over $\mathbb{A}$ and G ranges over $2^{\mathbb{A}}$.

We call the corresponding logic CALR. Let us compare its expressiveness to that of ATDEL_3, which is the version of ATDEL that respects the following condition.

(iii) for each agent $i \in \mathbb{A}$, the set of available actions for i is $\mathbb{T}_i = \{(\mathrm{K}_i\varphi, \emptyset) \mid \mathrm{K}_i\varphi \in \mathcal{L}_{\mathrm{EL}}\}$.

The key idea is to make ATDEL_3 respect three restrictions of CALR (which are already the case in CAL), namely, (1) public announcements are the only kind of action allowed, (2) each agent in $\mathbb{A}$ can only announce what is known and (3) only formulas in $\mathcal{L}_{\mathrm{EL}}$ can be announced by the agents in $\mathbb{A}$.

The translation function mltr from $\mathcal{L}_{\mathrm{CALR}}$ to the language of ATDEL_2 is defined as follows:

$$
\begin{array}{ll}
\mathrm{mltr}(\varphi) \stackrel{\mathrm{def}}{=} \varphi & (\text{if } \varphi \in \mathcal{L}_{\mathrm{EL}}) \\
\mathrm{mltr}([\varphi_1]\varphi_2) \stackrel{\mathrm{def}}{=} [\mathrm{mltr}'(\varphi_1) \cup \mathrm{skip}_{\mathbb{A}}]\varphi_2 & \\
\mathrm{mltr}(\langle\!\langle G \rangle\!\rangle \varphi) \stackrel{\mathrm{def}}{=} \langle\!\langle G \rangle\!\rangle \mathbf{X}\, \mathrm{mltr}(\varphi) &
\end{array}
$$

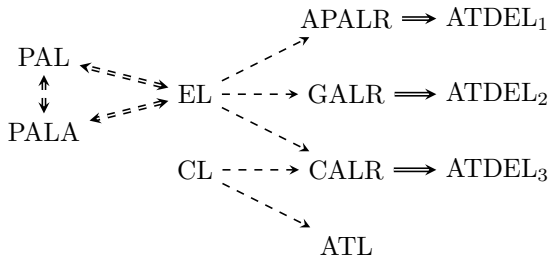


Figure 5.2 – Expressiveness relations

where mltr' is defined by:

$$\mathrm{mltr}'(\mathrm{K}_i\chi) \stackrel{\text{def}}{=} \{(i, (\chi, \emptyset))\}$$
$$\mathrm{mltr}'(\psi_1 \wedge \psi_2) \stackrel{\text{def}}{=} \mathrm{mltr}'(\psi_1) \cup \mathrm{mltr}'(\psi_2)$$

The intuition behind the second clause of mltr is that public announcements φ_1, which are always of the form $\bigwedge_{k\leq n} \mathrm{K}_{i_k}\psi_k$, are translated to $\bigcup_{k\leq n}\{(i_k, (\psi_k, \emptyset))\}$.

We obtain the following result.

Proposition 5.7. $\mathrm{CALR} \leq \mathrm{ATDEL}_3$.

5.4.5 Summary

The diagram in Figure 5.2 summarizes currently known expressiveness relations between some logics: a double arrow from logic L_1 to logic L_2 represents $\mathrm{L}_1 \leq \mathrm{L}_2$, a single arrow represents $\mathrm{L}_1 \leq \mathrm{L}_2$ but not $\mathrm{L}_1 = \mathrm{L}_2$, dashed arrows represent previously established results and black arrows represent new results. Some transitive and reflexive arrows are omitted.

5.5 The Next-fragment of ATDEL

The next-fragment of ATDEL has some interesting properties which are worth to be examined separately. We do so in this section.

5.5.1 Axiom System

Table 5.1 displays the axiom system of the next-fragment of ATDEL. Principles (CPL), ($\mathrm{KT5}_n$), (RMP) and (RNK) are the standard ones for epistemic logic. Principles (AA), (AN), (AC), (AK) and (RNA) are similar to the reduction axioms and rule of inference of public announcement logic (van Ditmarsch, van der Hoek, et al. 2007; Plaza 1989) and public announcement logic with assignment (van Ditmarsch et al. 2005). These principles follow directly from the semantics of ATDEL. Principle (AD) captures the

(CPL)	All axiom schemas of CPL	
(KT5$_n$)	All axiom schemas of KT5$_n$	
(AA)	$[e_{\mathbb{A}}]p \leftrightarrow (\mathrm{pre}(e_{\mathbb{A}}) \rightarrow \mathrm{post}(e_{\mathbb{A}})(p))$	(action and atoms)
(AN)	$[e_{\mathbb{A}}]\neg\varphi \leftrightarrow (\mathrm{pre}(e_{\mathbb{A}}) \rightarrow \neg[e_{\mathbb{A}}]\varphi)$	(action and negation)
(AC)	$[e_{\mathbb{A}}](\varphi \wedge \psi) \leftrightarrow ([e_{\mathbb{A}}]\varphi \wedge [e_{\mathbb{A}}]\psi)$	(action and conjunction)
(AK)	$[e_{\mathbb{A}}]\mathrm{K}_i\varphi \leftrightarrow (\mathrm{pre}(e_{\mathbb{A}}) \rightarrow \mathrm{K}_i[e_{\mathbb{A}}]\varphi)$	(action and knowledge)
(AD)	$[e_{\mathbb{A}}](\varphi \rightarrow \psi) \rightarrow ([e_{\mathbb{A}}]\varphi \rightarrow [e_{\mathbb{A}}]\psi)$	(action distribution)
(AS)	$([e_G]\varphi \wedge [f_H]\psi) \rightarrow [e_G \cup f_H](\varphi \wedge \psi)$ if $G \cap H = \emptyset$	(action superadditivity)
(AG)	$(\langle e_G\rangle\top \wedge [e_G]\varphi) \rightarrow \langle\!\langle G\rangle\!\rangle \mathbf{X}\varphi$	(action and group)
(RMP)	From φ and $\varphi \rightarrow \psi$ infer ψ	(modus ponens)
(RNK)	From φ infer $\mathrm{K}_i\varphi$	(knowledge necessitation)
(RNA)	From φ infer $[e_{\mathbb{A}}]\varphi$	(action necessitation)
(RDA)	From $\eta([e_G \cup f_H]\varphi)$ for all $f \in E$ infer $\eta([e_G]\varphi)$	(deriving action)
(RDG)	From $\eta([e_G]\bot \vee \langle e_G\rangle\varphi)$ for all $e \in E$ infer $\eta([G]\mathbf{X}\varphi)$	(deriving group)

Table 5.1 – ATDEL Next-fragment Axiom System

very natural intuition that, if $\varphi \to \psi$ is true after every possible occurrence of $e_{\mathbb{A}}$ then, if φ is true after $e_{\mathbb{A}}$ then so is ψ.

Principle (AS) is sometimes called 'superadditivity'. It captures the intuition that, if a group G enforces φ by executing the actions in e, and group H enforces ψ by executing the actions in f, then, by working together, the two groups enforce outcomes satisfying both φ and ψ.

Principle (RDA) captures the intuition that, if a group G enforces φ by executing the actions in e, in spite of what other agents do, in particular, in spite of what H does, then G enforces φ by executing the actions in e. The principles (AG) and (RDG) capture the intuition that, if group G enforces φ by executing action e, then G is able to enforce φ. Principles (RDA) and (RDG) use necessity forms. They have been inspired by rule $R^{\omega}(\Box)$ of APAL (cf. (Balbiani et al. 2008)). We need such rules to "encode" the quantification over actions embedded in the semantics the operator $[\ ]$. These inference rules work (so to say) as the counter-positives of Proposition 5.9.3 (which is shown using (AS) on page 98) and Axiom (AG), respectively. Also note that, unlike other logics in the DEL family, this axiom system does not permit a reduction to EL. Such reduction cannot be achieved, because ATDEL is at least as expressive as APALR (cf. Proposition 5.5), which has been shown to be strictly more expressive than EL in (Balbiani et al. 2008).

Theorem 5.8 (Soundness)**.** *All principles in Table 5.1 are valid in ATDEL.*

As usual, a formula $\varphi \in \mathcal{L}_{\mathrm{ATDEL}}$ is a theorem of ATDEL, noted $\vdash \varphi$, if and only if φ is an instantiation of some axiom from the axiom system of ATDEL, or it is generated by the application of some inference rule from the axiom system of ATDEL to theorems of ATDEL.

In the sequel, some interesting properties of ATDEL are derived. Some of them are used to prove completeness of the axiom system. Moreover, this exercise helps to illustrate how to correctly use the non-standard inference rules (RDA) and (RDG).

Proposition 5.9.

1. *if* $\vdash \varphi \leftrightarrow \psi$ *then* $\vdash \mathrm{K}_i\varphi \leftrightarrow \mathrm{K}_i\psi$ *(substitution of proved equivalences under* K_i*)*
2. *If* $\vdash \varphi$ *then* $\vdash [e_G]\varphi$ *(necessitation for* $[e_G]$*)*
3. $\vdash [e_G]\varphi \to [e_G \cup f_H]\varphi$ *(if* $G \cap H = \emptyset$*)* *(outcome monotonicity)*
4. $\vdash [e_G](\varphi \wedge \psi) \leftrightarrow ([e_G]\varphi \wedge [e_G]\psi)$ *(action and conjunction for* $[e_G]$*)*
5. $\vdash \mathrm{K}_i[e_G]\varphi \to [e_G]\mathrm{K}_i\varphi$ *(perfect recall)*
6. $\vdash [e_G](\varphi \to \psi) \to ([e_G]\varphi \to [e_G]\psi)$ *(action distribution for* $[e_G]$*)*
7. *If* $\vdash \varphi \to \psi$ *then* $\vdash [e_G]\varphi \to [e_G]\psi$ *(monotonicity of* $[e_G]$*)*
8. *If* $\vdash \varphi \leftrightarrow \psi$ *then* $\vdash [e_G]\varphi \leftrightarrow [e_G]\psi$ *(substitution of proved equivalences under* $[e_G]$*)*

Proof. We derive each property using the axiom system of ATDEL. Here we show only some of them.

2. 1. $\vdash \varphi$ (hypothesis)
   2. for all $f \in E$, $\vdash [e_G \cup f_{\overline{G}}]\varphi$ (from 1 with (RNA))
   3. $\vdash [e_G]\varphi$ (from 2 with (RDA), because $\sharp$ is a necessity form)

3. Assume $G \cap H = \emptyset$.
   1. $\vdash [f_H]\top$ (from Prop. 5.9.2)
   2. $\vdash ([e_G]\varphi \wedge [f_H]\top) \to [e_G \cup f_H](\varphi \wedge \top)$ (AS)
   3. $\vdash [e_G]\varphi \to [e_G \cup f_H]\varphi$ (from 1 and 2)

4. First, we derive the implication from the right to the left:
   1. for all $f \in E$, $\vdash ([e_G]\varphi \wedge [e_G]\psi) \to ([e_G \cup f_{\overline{G}}]\varphi \wedge [e_G \cup f_{\overline{G}}]\psi)$ (from Prop. 5.9.3)
   2. for all $f \in E$, $\vdash ([e_G]\varphi \wedge [e_G]\psi) \to [e_G \cup f_{\overline{G}}](\varphi \wedge \psi)$ (from 1 and (AC))
   3. $\vdash ([e_G]\varphi \wedge [e_G]\psi) \to [e_G](\varphi \wedge \psi)$ (from 2 with (RDA), because $([e_G]\varphi \wedge [e_G]\psi) \to \sharp$ is a necessity form)

   Now, we derive the implication from the left to the right:
   1. for all $f \in E$, $\vdash [e_G](\varphi \wedge \psi) \to [e_G \cup f_{\overline{G}}](\varphi \wedge \psi)$ (Prop. 5.9.3)
   2. for all $f \in E$, $\vdash [e_G \cup f_{\overline{G}}](\varphi \wedge \psi) \to ([e_G \cup f_{\overline{G}}]\varphi \wedge [e_G \cup f_{\overline{G}}]\psi)$ (from (AC))
   3. for all $f \in E$, $\vdash [e_G \cup f_{\overline{G}}](\varphi \wedge \psi) \to [e_G \cup f_{\overline{G}}]\varphi$ (from 2)
   4. for all $f \in E$, $\vdash [e_G](\varphi \wedge \psi) \to [e_G \cup f_{\overline{G}}]\varphi$ (from 1 and 3)
   5. $\vdash [e_G](\varphi \wedge \psi) \to [e_G]\varphi$ (from 4 with (RDA), because $[e_G](\varphi \wedge \psi) \to \sharp$ is a necessity form)

   Analogously, we obtain $\vdash [e_G](\varphi \wedge \psi) \to [e_G]\psi$. From this and 5, we obtain: $\vdash [e_G](\varphi \wedge \psi) \to ([e_G]\varphi \wedge [e_G]\psi)$. ∎

Propositions 5.9.2 and 5.9.6 together show that operators $[e_G]$ are normal modal operators. Proposition 5.9.5 corresponds to what is called 'perfect recall' in (Fagin et al. 1995). It is shown using superadditivity (AS), necessitation for knowledge (RNK), Axiom (K) for knowledge and Axiom knowledge and actions (AK). It captures the intuition that the knowledge of the agents either increases or remains the same after the occurrence of an event. This means that agents never loose information, i.e., once an agent knows something, this agent will never forget it. Together with the fact that each $R(i)$ is an equivalence relation, Proposition 5.9.5 implies that action occurrences are perceived by all agents, which implies that the agents also perceive the passage of time.

Proposition 5.10.

1. $\vdash \langle\!\langle G \rangle\!\rangle \mathbf{X}\top$ *(group activity)*
2. $\vdash \neg\langle\!\langle G \rangle\!\rangle \mathbf{X}\bot$ *(group non-blocking)*
3. $\vdash \neg\langle\!\langle \emptyset \rangle\!\rangle \mathbf{X}\neg\varphi \to \langle\!\langle \mathbb{A} \rangle\!\rangle \mathbf{X}\varphi$ *(joint determinism)*
4. $\vdash (\langle\!\langle G \rangle\!\rangle \mathbf{X}\varphi \wedge \langle\!\langle H \rangle\!\rangle \mathbf{X}\psi) \to \langle\!\langle G \cup H \rangle\!\rangle \mathbf{X}(\varphi \wedge \psi)$ *(if $G \cap H = \emptyset$)* *(group superadditivity)*
5. *If* $\vdash \varphi \to \psi$ *then* $\vdash \langle\!\langle G \rangle\!\rangle \mathbf{X}\varphi \to \langle\!\langle G \rangle\!\rangle \mathbf{X}\psi$ *(monotonicity of $\langle\!\langle G \rangle\!\rangle \mathbf{X}$)*

6. *If* $\vdash \varphi \leftrightarrow \psi$ *then* $\vdash \langle\!\langle G \rangle\!\rangle \mathbf{X}\varphi \leftrightarrow \langle\!\langle G \rangle\!\rangle \mathbf{X}\psi$ *(substitution of proved equivalences under* $\langle\!\langle G \rangle\!\rangle \mathbf{X}$*)*

Propositions 5.10.1–5 are the principles satisfied by operators $\langle G \rangle$ of coalition logic (CL).

The next theorem states completeness of the axiom system in Table 5.1.

Theorem 5.11 (Completeness)**.** *Every valid formula* $\varphi \in \mathcal{L}_{\mathrm{X}}$ *is a theorem in ATDEL.*

5.5.2 Decision Procedures

Algorithm 5.1 decides model checking for the next-fragment of all versions of ATDEL in which the sets of actions $\mathbb{T}_i$ are finite. Note that the set $\mathbb{T} = \{\mathbb{T}_i \mid i \in \mathbb{A}\}$ is given as an argument. The idea is simple: if $\varphi \in \mathcal{L}_{\mathrm{EL}}$ it calls an existing model checker for EL (Algorithm 3.1); otherwise, it it simply tries all possible events in E. We have the following result.

input: A finite set $\mathbb{T}$ of actions, an epistemic model $\langle M, w \rangle$ and a formula $\varphi \in \mathcal{L}_{\mathrm{X}}$.
output: **true** if $\langle M, w \rangle \vDash \varphi$, **false** otherwise

```
1  function nfmc( 𝕋, ⟨M, w⟩, φ)
2      if φ ∈ ℒ_EL then return mlmc(⟨M, w⟩, φ)
3      else if φ = [e_G]ψ then
4          forall f_Ḡ ∈ E do
5              M′ := M|(e_G ∪ f_Ḡ)
6              if not nfmc( 𝕋, ⟨M′, w⟩, ψ) then return false
7          return true
8      else if φ = ⟨⟨G⟩⟩Xψ then
9          non-deterministically choose e_G ∈ E
10         return nfmc( 𝕋, ⟨M, w⟩, ⟨e_G⟩⊤) and nfmc( 𝕋, ⟨M, w⟩, [e_G]ψ)
```

Algorithm 5.1: ATDEL Next-fragment Model Checking

Theorem 5.12. *Model checking in the next-fragment of* ATDEL *with finite sets of actions is in PSPACE.*

Ågotnes et al. (2010) proved that model checking in both GAL and APAL is in PSPACE. These can be seen as two versions of ATDEL with infinite sets $\mathbb{T}_i$ (cf., Section 5.4). Their algorithm is similar to Algorithm 5.1, but instead of choosing events in the set E, it computes definable restrictions of model M via bisimulation contractions. All these restrictions correspond to valid public announcements in the languages of APAL and GAL. However, it is not clear if all versions of ATDEL with infinite sets of actions could use the same algorithm. The difficulty is to provide a non-resource consuming method to verify whether some given definable restriction of M would, indeed, be a valid event from E.

Satisfiability checking in the next-fragment of ATDEL is not decidable in general. It follows immediately from the non-decidability of satisfiability checking in APAL (French and van Ditmarsch 2008), which can easily be transferred to APALR, and because there is a version of the next-fragment of ATDEL that is at least as expressive as APALR (cf. Proposition 5.5).

However, when the sets of available actions are finite, so is the set E of possible events. In this case, one can define an algorithm that decides satisfiability. Such algorithm depends on Principle substitution of proved equivalences (RSE), which soundness is proved below.

Lemma 5.13. *The following principle is sound in the next fragment of ATDEL:*

(RSE) *If* $\vdash \varphi \leftrightarrow \psi$ *then* $\vdash \chi \leftrightarrow \chi[\varphi/\psi]$ *(substitution of proved equivalences)*

Theorem 5.14. *Satisfiability checking in the next-fragment of ATDEL with finite sets of actions is decidable.*

When the sets of actions available for the agents in $\mathbb{A}$ is finite, the infinitary rules (RDA) and (RDG) can be replaced by the following two axioms:

$$\text{(RA}') \qquad \bigwedge_{f \in E} [e_G \cup f_{\overline{G}}]\varphi \to [e_G]\varphi$$

$$\text{(RG}') \qquad \bigwedge_{e \in E} ([e_G]\bot \vee \langle e_G\rangle\varphi) \to [\![G]\!]\mathbf{X}\varphi$$

Axiom (RA′) and Proposition 5.9.3 together imply the following reduction axiom:

$$[e_G]\varphi \leftrightarrow \bigwedge_{f \in E} [e_G \cup f_{\overline{G}}]\varphi$$

This means that successive applications of this equivalence and RSE (Lemma 5.13) replace operators $[e_G]$ for operators $[e'_{\mathbb{A}}]$, i.e., formulas containing actions executed by a group G can be replaced by formulas containing actions executed by the entire set of agents $\mathbb{A}$. Then, using Axioms (AA), (AN), (AC) and (AK), operators $[e'_{\mathbb{A}}]$ can be eliminated. The result is a (huge) formula in $\mathcal{L}_{\mathrm{EL}}$.

Similarly, axioms (RG′) and (AG) together imply the following reduction axiom:

$$\langle\!\langle G\rangle\!\rangle\mathbf{X}\varphi \leftrightarrow \bigvee_{e \in E} (\langle e_G\rangle\top \wedge [e_G]\varphi)$$

This means that operators $\langle\!\langle\ \rangle\!\rangle\mathbf{X}$ can be eliminated from formulas by successive applications of this equivalence and (RSE). Since, in this case, operators $[\ ]$ can also be eliminated, the next-fragment of ATDEL with a finite set of actions is reducible to epistemic logic.

This, all together, implies that the next-fragment of ATDEL with a finite set of actions is reducible to epistemic logic. Since satisfiability checking in EL is decidable, so is satisfiability checking in this fragment.

	All principles in Table 5.1 (of Page 96)	
(FPA)	$\langle\!\langle G\rangle\!\rangle \mathbf{A}\varphi \to (\varphi \wedge \langle\!\langle G\rangle\!\rangle \mathbf{X}\langle\!\langle G\rangle\!\rangle \mathbf{A}\varphi)$	(fixed-point for always)
(FPU)	$\langle\!\langle G\rangle\!\rangle (\psi \mathbf{U} \varphi) \leftrightarrow (\varphi \vee (\psi \wedge \langle\!\langle G\rangle\!\rangle \mathbf{X}\langle\!\langle G\rangle\!\rangle (\psi \mathbf{U} \varphi)))$	(fixed-point for until)
(RIA)	From $\chi \to (\varphi \wedge \langle\!\langle G\rangle\!\rangle \mathbf{X}\chi)$ infer $\chi \to \langle\!\langle G\rangle\!\rangle \mathbf{A}\varphi$	(induction for always)
(RIU)	From $(\varphi \vee (\psi \wedge \langle\!\langle G\rangle\!\rangle \mathbf{X}\chi)) \to \chi$ infer $\langle\!\langle G\rangle\!\rangle (\psi \mathbf{U} \varphi) \to \chi$	(induction for until)

Table 5.2 – ATDEL Axiom System (for a finite sets of actions)

Note that the algorithm induced by Theorem 5.14 is probably non-optimal. Let φ be given, one can imagine a tableaux-like method that explores the canonical epistemic model filtrated by φ branch by branch. Since the size of each branch is a polynomial function on the length of φ, this may lead to an algorithm that is less resource consuming. We leave such kind of improvement to future work though.

5.6 Full ATDEL

5.6.1 Axiom System

Table 5.2 displays the axiom system of full ATDEL. The principles therein are standard for logics containing operators always and until. For instance, they are analogous to the principles present in the axiom system of ATL, given, e.g., in (Goranko and van Drimmelen 2006). They are proved to be sound in Theorem 5.15 below.

Theorem 5.15 (Soundness)**.** *All principles in Table 5.2 are valid in ATDEL.*

However, to prove completeness, we need to add the assumption that all sets $\mathbb{T}_i$ are finite! This is so because the technique used (similar to the one used in (Halpern and Moses 1992) for the common knowledge operator) requires a finite canonical model. The complete axiom system for ATDEL with infinite sets $\mathbb{T}_i$ is left as an open question.

Theorem 5.16 (Completeness)**.** *Every formula $\varphi \in \mathcal{L}_{\text{ATDEL}}$ which is valid in ATDEL with finite sets of actions is a theorem of ATDEL.*

5.6.2 Decision Procedures

Algorithm 5.2 decides model checking in full ATDEL in which the sets of actions available for the agents in $\mathbb{A}$ are finite. We have the following result.

Theorem 5.17. *Model checking in full ATDEL is in EXPSPACE.*

We leave as future work the lower bound of model checking in full ATDEL with finite sets of actions.

Satisfiability checking in full ATDEL with finite sets of actions is decidable. This can be shown as follows.

input: A fine set $\mathbb{T}$ of actions, an epistemic model $\langle M, w\rangle$ and a formula $\varphi \in \mathcal{L}_{\text{ATDEL}}$.
output: **true** if $\langle M, w\rangle \vDash \varphi$, **false** otherwise

```
1  function atdelmc( 𝕋, ⟨M, w⟩, φ)
2      if φ ∈ ℒ_X then return nfmc( 𝕋, ⟨M, w⟩, φ)
3      else if φ = ⟪G⟫Aψ then
4          return atdelmc( 𝕋, ⟨M, w⟩, ψ) and fpamc( 𝕋, ⟨M, w⟩, ψ, G, ∅)
5      else if φ = ⟪G⟫(ψ1 U ψ2) then
6          return atdelmc( 𝕋, ⟨M, w⟩, ψ2)
7              or ( atdelmc( 𝕋, ⟨M, w⟩, ψ1)
8                  and fpumc( 𝕋, ⟨M, w⟩, ψ1, ψ2, G, ∅) )

9  function fpamc( 𝕋, ⟨M, w⟩, ψ, G, S)
10     if M ∈ S then return true
11     else
12         non-deterministically choose e_G ∈ E
13         forall f_Ḡ ∈ E do
14             M' := M|(e_G ∪ f_Ḡ)
15             if not ( atdelmc( 𝕋, ⟨M', w⟩, ψ)
16             or not fpamc( 𝕋, ⟨M', w⟩, ψ, G, S ∪ {M} )) then
17                 return false
18         return true

19 function fpumc(𝕋, ⟨M, w⟩, ψ1, ψ2, G, S)
20     if M ∈ S then return true
21     else
22         non-deterministically choose e_G ∈ E
23         forall f_Ḡ ∈ E do
24             M' := M|(e_G ∪ f_Ḡ)
25             if not atdelmc( 𝕋, ⟨M', w⟩, ψ2) and
26             not ( atdelmc( 𝕋, ⟨M', w⟩, ψ1) and
27             fpumc( 𝕋, ⟨M', w⟩, ψ1, ψ2, G, S ∪ {M}) ) then
28                 return false
29         return true
```

Algorithm 5.2: ATDEL Model checking

Theorem 5.18 (Finite Model Theorem for ATDEL)**.** *Every satisfiable formula* $\varphi \in \mathcal{L}_{\text{ATDEL}}$ *is satisfiable in a finite model.*

The proof is based on the fact that the filtrated canonical model for ATDEL is constructed using a finite set of formulas $\text{sub}(\varphi)$ (also see Section 3.6). This enables us to show the following theorem.

Theorem 5.19 (Decidability of ATDEl)**.** *Satisfiability checking in ATDEL with a finite number of actions is decidable.*

5.7 Related Work and Discussion

Apart from the formalisms mentioned in Section 5.4, there are some other containing actions by groups and group modalities in their languages that are worth to be mentioned here. They are the coalition action logic (Borgo 2007), the alternating-time temporal logic with explicit strategies (Walther et al. 2007), the dynamic logic of agency (DDL) (Herzig and Lorini 2010a) and the logic CEDL, presented on Chapter 4 The first two differ from ATDEL in several aspects. The most important of them is perhaps the fact that they do not model the knowledge of the agents and, thus, also do not have epistemic actions. The third and fourth ones model the knowledge of agents and their languages look very similar to that of ATDEL. But, their semantics is completely defined in terms of Kripke structures, instead of model updates. This difference is reflected on their axiom system. Both, DDL and CEDL do not validate reduction axioms as ATDEL does (i.e., Axioms (AA), (AN), (AC) and (AK)). One can argue that those formalisms are able to model "more actions", instead of only public announcements and actions with factual effects. This is indeed a limitation of ATDEL. But the two kinds of actions ATDEL is able to model are already very expressive. The examples on Section 5.3 give a strong support for this argument. Second, as mentioned before, because of its action descriptions, the specifications of scenarios in ATDEL are generally more succinct than in DDL and CEDL. This is the case because one must deal with the so-called frame axioms in DDL and CEDL.

5.8 Conclusion

In this chapter, we saw the alternating-time temporal dynamic epistemic logic, wherein one may write formulas of the form $\langle\!\langle G \rangle\!\rangle \mathbf{X}\varphi$, which mean 'there is an action by group G after which φ is true, in spite of what the other agents do' (as well as always and until versions of it). Differently from the previously existent coalition announcement logic, ATDEL also contains actions with factual effects, temporal operators, and is equipped with a sound and complete axiom system.

We saw that ATDEL subsumes several logics of the dynamic epistemic logics family. It is also shown to be useful in modeling multi-agent systems through examples of collaborative and competitive scenarios. Moreover, compared to other multi-agent logics, such as ATL and STIT, it adds the possibility to model agents' knowledge and it has

concise descriptions of actions, Thus, permitting reasonable sized multi-agent systems specifications.

In the next chapter, we will leave knowledge and actions behind and discuss agents belief and belief change instead. Belief change is more complicated than knowledge change because, apart from learning, agents may also have to revise their beliefs. This is subject of a whole different area, called belief revision theory.

Chapter 6

Belief Change in Multi-agent Settings

In the last two chapters, we designed modal logics for reasoning about actions and knowledge in multi-agent environments. In order to do that, the formalisms incorporate some dynamic operators as well the standard epistemic logic of knowledge, i.e., $KT5_n$. Agents are able to execute various actions. In particular, they can communicate their knowledge. In ATDEL, the communication action available for the agents is the public announcement. This is an action that gives information to all agents in the environment. Just after a public announcement, the contents of the announcement is common knowledge among all agents. In some applications though, agents knowledge can be inaccurate. By communicating with others, agents may confirm the information they have, or realise that they should revise their knowledge base. For these applications, we have to incorporate an epistemic logic of belief, instead of knowledge.

However, the replacement of $KT5_n$ for the standard epistemic logic of belief, $KD45_n$, in ATDEL does not work. This is so because the axiom (AK) on Table 5.1 would not be valid. The explanation is somewhat technical and can be found in (Balbiani et al. 2012). The solution we proposed on that work is only palliative: when an agent receives an information that contradicts her current beliefs, the agent just ignores it! Therefore, if we really want to be able to deal with scenarios where agents may have inaccurate information, we need to enter the realm of belief revision theory.

In this chapter, we make one step on that direction. We design operators that change the beliefs of the agents in $KD45_n$ models.

Alchourrón, Gärdenfors and Makinson (Alchourrón et al. 1985; Gärdenfors 2008) proposed postulates for the expansion, contraction and revision of belief sets. These postulates logically encode the constraints expected on the behaviour of such operators. Several representation theorems in terms of maximal consistent sets (Alchourrón et al. 1985), plausibility relations on formulas (Gärdenfors 2008), or plausibility relations on worlds exist (Katsuno and Mendelzon 1991b), allowing to define operators with these

expected properties. But, in the case of $\mathrm{KD45}_n$ models, this task is more complicated than in the standard AGM framework, because, in a multi-agent context, a new piece of evidence can take different forms. For instance, it can be observed, transmitted, or available to every agent or only to some of them.

Here, we consider private and group change, i.e., one or more agents receive some new piece of information, and then we look at defining the new $\mathrm{KD45}_n$ model that represents the new epistemic situation. We consider only objective pieces of information, i.e., information about the environment (world). The problem of considering change by subjective information, i.e., information about the beliefs of other agents, is more difficult and is left for future work. We study both expansion and revision. For each case, we provide a translation of AGM postulates for the multi-agent setting, and some specific operators.

The remainder of the chapter synthesises the results published in (Caridroit et al. 2015a,b), with some small additions, and is organised as follows. After some considerations about multi-agent belief sets, we study private expansion. In Section 6.2, we translate the AGM postulates for this kind of scenario and then propose a particular expansion operator that satisfies these postulates. In Section 6.3, we translate the AGM postulates for private revision, and then propose a family of revision operators. We discuss some related work in Section 6.4 and then conclude the chapter in Section 6.5.

6.1 Multi-agent Belief Sets

We are interested here in a framework with several agents, each of them having her own beliefs about the state of the world and about the beliefs of other agents. Therefore, we take the epistemic logic commonly used for modelling agents beliefs in multi-agent systems, i.e., $\mathrm{KD45}_n$ (see Section 3.7.1). Here, a pointed $\mathrm{KD45}_n$ model represents a set of $|\mathbb{A}|$ belief sets, one for each $i \in \mathbb{A}$. Formally, we have the following.

Definition 6.1 (Multi-agent Belief Set)**.** Let a pointed $\mathrm{KD45}_n$ Kripke model be given. The belief set of agent $i \in \mathbb{A}$ is the set $K_i^{\langle M,w\rangle} = \{\varphi \mid \langle M, w\rangle \vDash \mathrm{B}_i\varphi\}$.

In other words, the belief set of an agent is the set of formulas that the agent believes in the given model. Note that $K_i^{\langle M,w\rangle}$ is a deductively closed set.

We also define the objective belief set of agent i, (i.e., what i believes about the state of the world). This is the set $O_i^{\langle M,w\rangle} = K_i^{\langle M,w\rangle} \cap \mathcal{L}_{\mathrm{CPL}}$. Note that objective belief sets are deductively closed subsets of $\mathcal{L}_{\mathrm{CPL}}$. Therefore, we have that objective belief sets are AGM belief sets.

In the following, we make the assumption that the new information is a consistent formula. Making a change by an inconsistent formula is allowed by AGM postulates, but it is not of much interest in practical applications. Furthermore, recall from Section 3.4.1 that axiom (D) forbids inconsistent beliefs.

(E $+_a$ 0)	$V'(w') = V(w)$	(no factual change)
(E $+_a$ 1)	If $\langle M, w\rangle \not\models \mathrm{B}_a\neg\varphi$ then $\langle M, w\rangle +_a \varphi \in \mathrm{KD45}_n$	(closure)
(E $+_a$ 2)	$\langle M, w\rangle +_a \varphi \models \mathrm{B}_a\varphi$	(success)
(E $+_a$ 3)	$\langle M, w\rangle \models \mathrm{B}_i\psi$ iff $\langle M, w\rangle +_a \varphi \models \mathrm{B}_i\psi$, for $i \neq a$	(privacy)
(E $+_a$ 4)	If $\langle M, w\rangle \not\models \mathrm{B}_a\neg\varphi$ then $\langle M, w\rangle \models \mathrm{B}_a^n\mathrm{B}_i\psi$ iff $\langle M, w\rangle +_a \varphi \models \mathrm{B}_a^n\mathrm{B}_i\psi$ for $a \neq i$ and $n \geq 1$	(believed privacy)
(E $+_a$ 5)	If $\langle M, w\rangle \models \mathrm{B}_a\psi$ then $\langle M, w\rangle +_a \varphi \models \mathrm{B}_a\psi$	(inclusion)
(E $+_a$ 6)	If $\langle M, w\rangle \models \mathrm{B}_a\varphi$ then $(\langle M, w\rangle +_a \varphi) \leftrightarrows \langle M, w\rangle$	(vacuity)
(E $+_a$ 7)	If $\langle M_1, w_1\rangle \models \mathrm{B}_a\psi$ implies $\langle M_2, w_2\rangle \models \mathrm{B}_a\psi$ then $\langle M_1, w_1\rangle +_a \varphi \models \mathrm{B}_i\chi$ implies $\langle M_2, w_2\rangle +_a \varphi \models \mathrm{B}_i\chi$	(monotonicity)
(E $+_a$ 8)	For all $\langle M', w'\rangle$, if $\langle M', w'\rangle$ satisfies (E $+_a$ 1)–(E $+_a$ 7) then $\langle M, w\rangle +_a \varphi \models \mathrm{B}_a\psi$ implies $\langle M', w'\rangle \models \mathrm{B}_a\psi$	(minimality)

Table 6.1 – AGM postulates for private expansion in $\mathrm{KD45}_n$

6.2 Private Expansion

6.2.1 Private Expansion Postulates

In this section, we focus on private expansion, i.e., only one agent increases her beliefs. Beliefs of other agents, as well as the higher order beliefs, remain unchanged. Hereafter, we note a the agent that performs the belief change.

The result of the private expansion of a pointed Kripke model $\langle M, w\rangle$ by $\varphi \in \mathcal{L}_{\mathrm{CPL}}$ for agent a is a new pointed Kripke model $\langle M, w\rangle +_a \varphi$. The AGM postulates for expansion can be rewritten for $\mathrm{KD45}_n$ models, as depicted in Table 6.1.

Postulate (E $+_a$ 0) stipulates that the actual world does not change. As usual in belief revision, the state of the world does not change[1], only the agents beliefs are allowed to change. Postulate (E $+_a$ 1) stipulates that, if the new information does not contradict agent a beliefs, then the resulting model remains a $\mathrm{KD45}_n$ model after the private expansion. Indeed, when the expansion is done by a formula that contradicts agent a beliefs, the result infringes axiom (D) for agent a. The model is therefore no longer in $\mathrm{KD45}_n$. Therefore, this postulate guarantees that $K_a^{\langle M,w\rangle} +_a \varphi$ is a belief set. Postulage (E $+_a$ 2) is the success postulate. It stipulates that, after the private expansion by φ, agent a believes that φ. It guarantees that $\varphi \in K_a^{\langle M,w\rangle} +_a \varphi$. Postulate (E $+_a$ 3) stipulates that the beliefs of all other agents (different from a) do not change. This means that $K_i^{\langle M,w\rangle} +_a \varphi = K_i^{\langle M,w\rangle}$, for $i \neq a$. Postulate (E $+_a$ 4) stipulates that agent a beliefs about other agents beliefs do not change. Postulates (E $+_a$ 5) and

[1] When the state of the world evolves, one has to make an update (Herzig et al. 2005; Katsuno and Mendelzon 1991a).

(E $+_a$ 6) ensure that, if φ is already believed by agent a, then the private expansion does not change anything, so the resulting model is bisimilar to the initial one. These postulates guarantee, that $\psi \in K_a^{\langle M,w\rangle}$ implies $\psi \in K_a^{\langle M,w\rangle} +_a \varphi$ and that $\varphi \in K_a^{\langle M,w\rangle}$ implies $K_a^{\langle M,w\rangle} = K_a^{\langle M,w\rangle} +_a \varphi$. Postulate (E $+_a$ 7) is the translation of the monotonicity property. It stipulates that, if a model allows more inferences than another one, then the expansion of the first one allows more inferences than the expansion of the second one. Finally, postulate (E $+_a$ 8) is the minimality postulate. It stipulates that the result of the expansion of the model by φ is a minimal belief change.

One can prove the following two propositions.

Proposition 6.1. *Let $+_a$ be an expansion operator satisfying postulates* (E $+_a$ 0)–(E $+_a$ 8)*. The operator $+$ defined by $O_a^{\langle M,w\rangle} + \varphi = O_a^{\langle M,w\rangle} +_a \varphi$ is an AGM expansion operator, i.e., it satisfies postulates* (K + 1)–(K + 6) *in Section 2.2.*

Proposition 6.2. *There is a unique (up to model bisimilarity) private expansion operator satisfying* (E $+_a$ 0)–(E $+_a$ 8)*.*

This means that the postulates in Table 6.1 are a conservative extension of the usual AGM expansion ones.

6.2.2 A Private Expansion Operator

Now, we give a constructive definition of the private expansion operator characterized in the previous section.

Hereafter, we use v_w^x to denote a *copy* of the possible world w, but having the Boolean valuation x.

Definition 6.2 (Expansion of $\langle M, w_0\rangle$ by φ for agent a.)**.** Let $\langle M, w_0\rangle = \langle W, R, V, w_0\rangle$ be a KD45$_n$ pointed model, and φ be a consistent objective formula (i.e., $\varphi \in \mathcal{L}_{\text{CPL}}$ and $\not\models \neg\varphi$). The private expansion of $\langle M, w_0\rangle$ by φ for agent a is $\langle M, w_0\rangle +_a \varphi = \langle W', R', V', w_0'\rangle$, such that:

- $W' = \{w_0'\} \cup W \cup W^\varphi$, where
  - $W^\varphi = \bigcup_{w \in R_a(w_0)} W_w^\varphi$
  - $W_w^\varphi = \bigcup_{x \in X} \{v_w^x\}$
  - $X = \{V(w) \mid w \in R_a(w_0) \cap [\![\varphi]\!]_M\}$
- $R_a' = R_a \cup R_a^\varphi \cup R_a^0$, where
  - $R_a^\varphi = \{(w_1^\varphi, w_2^\varphi) \mid w_1^\varphi, w_2^\varphi \in W^\varphi\}$
  - $R_a^0 = \{(w_0', w^\varphi) \mid w^\varphi \in W^\varphi\}$
- $R_i' = R_i \cup R_i^{\overrightarrow{\varphi}} \cup R_i^0$, for $i \neq a$, where
  - $R_i^{\overrightarrow{\varphi}} = \{(v_w^x, w') \mid i(w, w') \in R_i$ and $v_w^x \in W^\varphi\}$, for $i \neq a$

- $R_i^0 = \{(w_0', w) \mid (w_0, w) \in R_i\}$, for $i \neq a$

- $V'(w) = V(w)$, for $w \in W$

- $V'(v_w^x) = x$, for $v_w^x \in W^\varphi$

- $V'(w_0') = V(w_0)$

When the agent a expands her beliefs, the model must change in order to represent these new beliefs, but other agents beliefs should remain unchanged. The new set of possible worlds W' contains all possible worlds of the initial model plus a new real world w_0' and a set of worlds W^φ representing the new beliefs of agent a. The set W^φ contains a copy of each world in $R_a(w_0)$ which does not contradict φ.

The new accessibility relation R_a' contains the initial relation R_a and the sets R_a^φ and R_a^0. The set R_a^0 consists of pairs (w_0', w^φ) where $w^\varphi \in W^\varphi$, thus modifying the beliefs of the agent performing the expansion. The set R_a^φ consists of pairs $(w_1^\varphi, w_2^\varphi) \in W^\varphi$. The worlds in W^φ thus form a clique, because they are equally plausible for the agent performing the expansion.

Each accessibility relation R_i', for $i \neq a$, contains the initial relation R_i and the sets R_i^0 and $R_i^{\overrightarrow{\varphi}}$. The set R_i^0 consists of all pairs (w_0', w) such that $(w_0, w) \in R_i$, thus preserving the beliefs of agents not performing expansion and higher order beliefs of all agents. The set $R_i^{\overrightarrow{\varphi}}$ consists of pairs (v_w^x, w'), where $v_w^x \in W^\varphi$ such that $(w, w') \in R_i$, thus keeping higher-order beliefs of the agent performing the expansion.

We can now show that:

Proposition 6.3. *The operator* $+$ *satisfies* $(\mathrm{E} +_a 0)$–$(\mathrm{E} +_a 8)$.

As a direct consequence of Proposition 6.2, we know that this operator is the unique private expansion operator.

Example 6.1. Consider the $\mathrm{KD45}_n$ model $\langle M, w_0 \rangle$ depicted in Figure 6.1. In this situation, agent 1 believes $\neg p$ and she believes that agent 2 also believes $\neg p$. Agent 2 believes $\neg p \wedge \neg q$, and she believes that agent 1 believes $\neg p$. After the expansion by q, agent 1 believes $\neg p \wedge q$. The model $\langle M', w_0' \rangle$ obtained after the expansion is also depicted in Figure 6.1. The world having the valuation $\neg p \wedge q$ has to be duplicated in order to keep the higher-order beliefs of agent 1. The beliefs of the agent 2 remain unchanged. In particular she still believes that agent 1 believes $\neg p$. ∎

Let us now show that our approach to private expansion can be defined using a product by a specific kind of pointed event model. This result is quite similar to what is shown by Aucher (2012) for internal models. Indeed, the expansion of Definition 6.2 is equivalent to a specific model update (see Definition 3.19). More precisely, $\langle M, w_0 \rangle +_a \varphi$

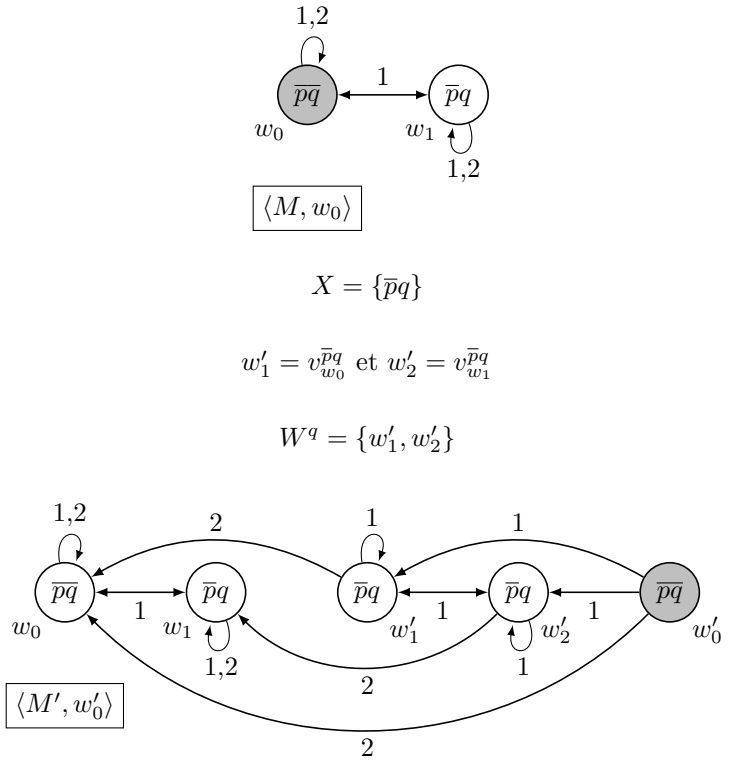


Figure 6.1 – Illustration of a private expansion. On the top, the initial model $\langle M, w_0 \rangle$. On the middle, the sets X and W^q (Definition 6.2). The model on the bottom corresponds to the expansion by q for agent 1, i.e., $\langle M, w_0 \rangle +_1 q$.

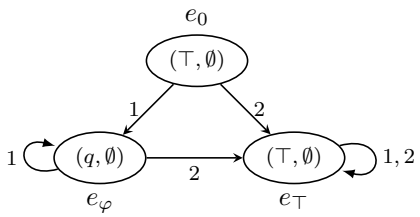


Figure 6.2 – Event model for the private expansion by q for 1

and $\langle M.N^{+_a}, w_0.e_0\rangle$ are bisimilar, where:

$$\begin{aligned} N^{+_a} &= \langle E, P, \mathrm{pre}, \mathrm{post}\rangle \\ E &= \{e_0, e_\varphi, e_\top\} \\ P_a &= \{(e_0, e_\varphi), (e_\varphi, e_\varphi), (e_\top, e_\top)\} \\ P_i &= \{(e_0, e_\top), (e_\varphi, e_\top), (e_\top, e_\top)\}, \text{ for all } i \neq a \\ \mathrm{pre}(e_\varphi) &= \varphi \\ \mathrm{pre}(e_0) &= \mathrm{pre}(e_\top) = \top \\ \mathrm{post}(e_0) &= \mathrm{post}(e_\varphi) = \mathrm{post}(e_\top) = \emptyset \end{aligned}$$

Proposition 6.4. $(\langle M, w_0\rangle +_a \varphi) \leftrightarrows \langle M.N^{+_a}, w_0.e_0\rangle$.

For example, the event model that corresponds to the private expansion for q by 1 in Example 6.1 is depicted in Figure 6.2. The reader may verify that the update of the pointed Kripke model $\langle M, w_0\rangle$ in Figure 6.1 by event e_0 in Figure 6.2 is bisimilar to the pointed Kripke model $\langle M', w_0'\rangle$ in Figure 6.1.

6.2.3 A General Expansion Operator

An interesting question is whether the private expansion operator can be generalised to groups of agents. To answer this question, we first need to make more precise what we mean by generalisation. Assume that a group of agents $G \subseteq \mathbb{A}$ receives a new piece of information φ, there are several possibilities:

Multiple Expansion: Each agent of the group makes a private expansion by φ. As a result, φ is believed by each agent in the group, but higher order believes do not change.

Common Expansion: All agents in the group make a public expansion by φ. As a result, φ is commonly believed among the agents in the group, but beliefs about the agents outside the group do not change.

Mixed Expansion: A mix of both kinds of expansions above.

It turns out that all three kinds of expansion are possible. Multiple expansion is simple. It amounts to make one private expansion for each member of group G. Common

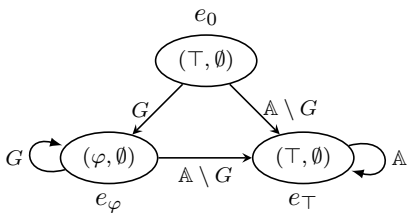


Figure 6.3 – Common expansion event model

expansion is also not at all hard to define. It amounts to a straightforward generalisation of private expansion. For simplicity, we do this using event models.

Definition 6.3 (Common Expansion Event Model)**.** Let $\langle M, w_0 \rangle$ be a KD45$_n$ pointed Kripke model, let φ be a consistent formula and $G \subseteq \mathbb{A}$. The common expansion of $\langle M, w_0 \rangle$ by φ for group G is $\langle M, w_0 \rangle +_G \varphi = \langle M.N^{+_G}, w_0.e_0 \rangle$, where:

$$
\begin{aligned}
N^{+_a} &= \langle E, P, \mathrm{pre}, \mathrm{post} \rangle \\
E &= \{e_0, e_\varphi, e_\top\} \\
P_a &= \{(e_0, e_\varphi), (e_\varphi, e_\varphi), (e_\top, e_\top)\}, \text{ for all } a \in G \\
P_i &= \{(e_0, e_\top), (e_\varphi, e_\top), (e_\top, e_\top)\}, \text{ for all } i \notin G \\
\mathrm{pre}(e_\varphi) &= \varphi \\
\mathrm{pre}(e_0) &= \mathrm{pre}(e_\top) = \top \\
\mathrm{post}(e_0) &= \mathrm{post}(e_\varphi) = \mathrm{post}(e_\top) = \emptyset
\end{aligned}
$$

The only difference between the event model for private expansion we saw before and the event model in Definition 6.3 above is the replacement of a for G. Figure 6.3 schematically illustrates this operator. When e_0 occurs, the agents performing the expansion believe that event e_φ is occurring, while the other agents believe that event $e_\top$ is occurring. In both events e_0 and $e_\top$ nothing changes. In e_φ, only the beliefs of agents in G changes. Moreover, the higher beliefs of agents in G do not change, because they all believe that the agents not in G believe that $e_\top$ is occurring.

Note that formula φ in Definition 6.3 can be a subjective formula. In fact, the common expansion operator defined above is a completely general expansion operator.

We have the following result.

Proposition 6.5. *Let $\langle M, w_0 \rangle$, φ and G be as in Definition 6.3. For each $a \in G$, we have $O_a^{\langle M, w_0 \rangle +_G \varphi} = O_a^{\langle M, w_0 \rangle +_a \varphi}$.*

Proposition 6.5 above means that operator $+_G$ is an AGM expansion for each agent $a \in G$. In particular, it means that postulates (E $+_a$ 0)–(E $+_a$ 8) are satisfied for each $a \in G$ and $i \notin G$. In fact, the common expansion operator should satisfy a stronger versions of the postulates in Table 6.1.

(R $\star_a$ 0)	$V'(w') = V(w)$	(no factual change)
(R $\star_a$ 1)	$\langle M, w\rangle \star_a \varphi \in \mathrm{KD45}_n$	(closure)
(R $\star_a$ 2)	$\langle M, w\rangle \star_a \varphi \vDash \mathrm{B}_a\varphi$	(success)
(R $\star_a$ 3)	$\langle M, w\rangle \vDash \mathrm{B}_i\psi$ iff $\langle M, w\rangle \star_a \varphi \vDash \mathrm{B}_i\psi$, for $i \neq a$	(privacy)
(R $\star_a$ 4)	$\langle M, w\rangle \vDash \mathrm{B}_a^k\mathrm{B}_i\psi$ iff $\langle M, w\rangle \star_a \varphi \vDash \mathrm{B}_a^k\mathrm{B}_i\psi$ for $i \neq a$ and $k \geq 1$	(believed privacy)
(R $\star_a$ 5)	If $\langle M, w\rangle \star_a \varphi \vDash \mathrm{B}_i\psi$ then $\langle M, w\rangle +_a \varphi \vDash \mathrm{B}_i\psi$	(inclusion)
(R $\star_a$ 6)	If $\langle M, w\rangle \nvDash \mathrm{B}_a\neg\varphi$ then $\langle M, w\rangle +_a \varphi \leftrightarrows \langle M, w\rangle \star_a \varphi$	(vacuity)
(R $\star_a$ 7)	If $\langle M^1, w^1\rangle \leftrightarrows \langle M^2, w^2\rangle$ and $\vDash \varphi \leftrightarrow \psi$ then $\langle M^1, w^1\rangle \star_a \varphi \leftrightarrows \langle M^2, w^2\rangle \star_a \psi$	(extensionality)
(R $\star_a$ 8)	If $\langle M, w\rangle \star_a (\varphi \wedge \psi) \vDash \mathrm{B}_i\chi$ then $(\langle M, w\rangle \star_a \varphi) +_a \psi \vDash \mathrm{B}_i\chi$	(iterated inclusion)
(R $\star_a$ 9)	If $\langle M, w\rangle \star_a \varphi \nvDash \mathrm{B}_a\neg\psi$ then $(\langle M, w\rangle \star_a \varphi) +_a \psi \vDash \mathrm{B}_i\chi$ implies $\langle M, w\rangle \star_a (\varphi \wedge \psi) \vDash \mathrm{B}_i\chi$	(iterated vacuity)

Table 6.2 – AGM postulates for private revision in $\mathrm{KD45}_n$

Conjecture 6.6. *The common expansion operator in Definition 6.3 satisfy postulates* (E $+_a$ 0)–(E $+_a$ 8)*, where the operators* B_a *are replaced for common belief operators* CB_G*.*

The third possibility of expansion amounts to a sequence of expansions. One private expansion for each agent that expands privately and one common expansion for each group that expands commonly.

6.3 Private Revision

6.3.1 Private Revision Postulates

Private revision behaves as expansion when there is no inconsistency between the agents beliefs and the new piece of evidence, but, unlike expansion, do not trivialize when this is not the case.

We denote the result of the private revision of the model $\langle M, w\rangle$ by a consistent CPL formula φ for agent a by the model $\langle M, w\rangle \star_a \varphi = \langle M', w'\rangle = \langle W', R', V', w'\rangle$. The AGM postulates for revision can be rewritten as in Table 6.2.

As for expansion, (R $\star_a$ 0) stipulates that there is no change on the actual world. (R $\star_a$ 1) stipulates that the model obtained after a revision is a $\mathrm{KD45}_n$ model. It also guarantees that $K_a^{\langle M,w\rangle}\star_a$ is a belief set. (R $\star_a$ 2) stipulates that φ is believed by a after the revision. Thus, we have that $\varphi \in K_a^{\langle M,w\rangle}$. (R $\star_a$ 3) stipulates that the beliefs of all

agents except a do not change. (R $\star_a$ 4) stipulates that the beliefs of the agent a about other agents do not change. (R $\star_a$ 5) and (R $\star_a$ 6) stipulate that, when the new piece of evidence is consistent with the beliefs of the agent, revision is just expansion. (R $\star_a$ 7) is an irrelevance of syntax postulate, stipulating that, if two formulas are logically equivalent, then they lead to the same revision results. (R $\star_a$ 8) and (R $\star_a$ 9) stipulate when the revision by a conjunction can be obtained by a revision followed by an expansion.

The revision operators we define are a conservative extension of usual AGM belief revision operators.

Proposition 6.7. *Let $\star_a$ be an revision operator satisfying postulates* (R $\star_a$ 0)–(R $\star_a$ 9). *The operator $*$ defined as $O_a^{\langle M,w\rangle} * \varphi = O_a^{\langle M,w\rangle \star_a \varphi}$ is an AGM revision operator (i.e., it satisfies* (K $*$ 1)–(K $*$ 8) *in Section 2.2).*

6.3.2 A Family Of Private Revision Operators

Let us now define a family of private revision operators. These operators are defined similarly to the expansion operator of the previous section, but in the cases when the new piece of evidence is inconsistent with the current beliefs of the agent they use a classical AGM belief revision operator $*$ in order to compute the new beliefs of the agent.

Definition 6.4 (Revision of $\langle M, w_0\rangle$ by φ for agent a)**.** Let $\langle M, w_0\rangle = \langle W, R, V, w_0\rangle$ be a KD45$_n$ model, let φ be a consistent CPL formula (i.e., $\varphi \in \mathcal{L}_{\text{CPL}}$ and $\not\models \neg\varphi$), and let $*$ be an AGM revision operator. We define the private revision of $\langle M, w_0\rangle$ by φ for agent a (with revision operator $*$) as $\langle M, w_0\rangle \star_a^* \varphi = \langle W', R', V', w_0'\rangle$, such that:

- if $R_a(w_0) \cap [\![\varphi]\!]_M \neq \emptyset$
  - then $X = \{V(w) \mid w \in R_a(w_0) \cap [\![\varphi]\!]_M\}$
  - else $X = \{x \mid x \subseteq \mathbb{P} \text{ and } x \vDash O_a^{\langle M,w_0\rangle} * \varphi\}$
- $W' = W \cup W^\varphi \cup \{w_0'\}$, where:
  - $W^\varphi = \bigcup_{w \in R_a(w_0)} W_w^\varphi$
  - $W_w^\varphi = \bigcup_{x \in X} \{v_w^x\}$
- $R_a' = R_a \cup R_a^\varphi \cup R_a^0$, where:
  - $R_a^\varphi = \{(w_1^\varphi, w_2^\varphi) \mid w_1^\varphi, w_2^\varphi \in W^\varphi\}$
  - $R_a^0 = \{(w_0', w^\varphi) \mid w^\varphi \in W^\varphi\}$
- $R_i' = R_i \cup R_i^{\overrightarrow{\varphi}} \cup R_i^0$ for $i \neq a$, where:
  - $R_i^{\overrightarrow{\varphi}} = \{(v_w^e, w') \mid (w, w') \in R_i, v_w^e \in W^\varphi\}$, for $i \neq a$
  - $R_i^0 = \{(w_0', w) \mid (w_0, w) \in R_i\}$, for $i \neq a$
- $V'(w) = V(w)$, for $w \in W$

- $V'(v^x_w) = x$, for $v^x_w \in W^\varphi$
- $V'(w'_0) = V(w_0)$

The construction of the revised model is similar to the construction of the expanded model discussed earlier. Only the new set of worlds W^φ is different: if the new information φ is considered possible by agent i, she performs an expansion, otherwise, each of the worlds of the new set W^φ has as valuation a (propositional) model of the new information φ.

The next result shows that these operators exhibit good logical properties.

Proposition 6.8. *The operators $\star^*_i$ satisfy* (R $\star_a$ 0)–(R $\star_a$ 9).

Example 6.2. We consider the model $\langle M, w_0\rangle$ in Figure 6.4, where agent 1 believes $\neg p \wedge \neg q$ and believes that agent 2 believes $p \wedge q$. Agent 2 believes $p \wedge q$ and believes that agent 1 believes $p \leftrightarrow q$. After the revision by $p \wedge q$, agent 1 believes $p \wedge q$, whereas the beliefs of agent 2 remain unchanged. The obtained model $\langle M', w'_0\rangle$ is also depicted in Figure 6.4. In this example, agent 1 uses Dalal's AGM revision operator $*_D$ (Katsuno and Mendelzon 1991b). We can observed that the revised model obtained using Definition 6.4 may not be minimal, in the sense that there may exist a smaller bisimilar model. Nevertheless, a minimal model can be obtained via bisimulation contraction. Here, this leads to the model $\langle M'', w'_0\rangle$, depicted in the same figure. ∎

Similarly to expansion, we can also define a model update that performs the revision.

Definition 6.5 (Private Revision Event Model)**.** Let $\langle M, w_0\rangle$ be a KD45$_n$ pointed Kripke model, let $\varphi \in \mathcal{L}_{\text{CPL}}$ such that $\not\models \neg\varphi$ and let $*$ be an AGM revision operator. The revision of $\langle M, w_0\rangle$ by φ for i is $\langle M, w_0\rangle.\langle N^{\star_a}, e_0\rangle$, where

$$
\begin{aligned}
N^{\star_a} &= \langle E, P, \text{pre}, \text{post}\rangle \\
E &= \{e_0, e_\top\} \cup \{e^x_w \mid v^x_w \in W^\varphi\} \\
P_a &= \{(e_0, e^x_w) \mid v^x_w \in W^\varphi\} \cup \{(e^{x_1}_{w_1}, e^{x_2}_{w_2}) \mid v^{x_1}_{w_1}, v^{x_2}_{w_2} \in W^\varphi\} \cup \{(e_\top, e_\top)\} \\
P_i &= \{(e_0, e_\top), (e_\top, e_\top)\} \cup \{(e^x_w, e_\top) \mid v^x_w \in W^\varphi\}, \text{ for } i \neq a \\
\text{pre}(e_0) &= \text{pre}(e_\top) = \top \\
\text{pre}(e^x_w) &= \bigwedge_{p \in V(w)} p \wedge \bigwedge_{p \in \mathbb{P} \setminus V(w)} \neg p \\
\text{post}(e_0) &= \text{post}(e_\top) = \emptyset \\
\text{post}(e^x_w)(p) &= \begin{cases} \top, & \text{if } x \vDash p \\ \bot, & \text{if } x \nvDash p \end{cases}
\end{aligned}
$$

where W^φ is as in Definition 6.4.

The event model in Definition 6.5 is similar to the one for expansion. The difference is that the possible event e_φ is replaced for a clique of possible events e^x_w. When e_0 occurs, agent a believes that one of the events e^x_w is occurring. As a result of the update, each possible world $w \in R_a(w_0)$ will be replaced by a set of possible worlds $w.e^x_w$, where $x \in X$, with a new valuation that satisfies φ in the resulting model.

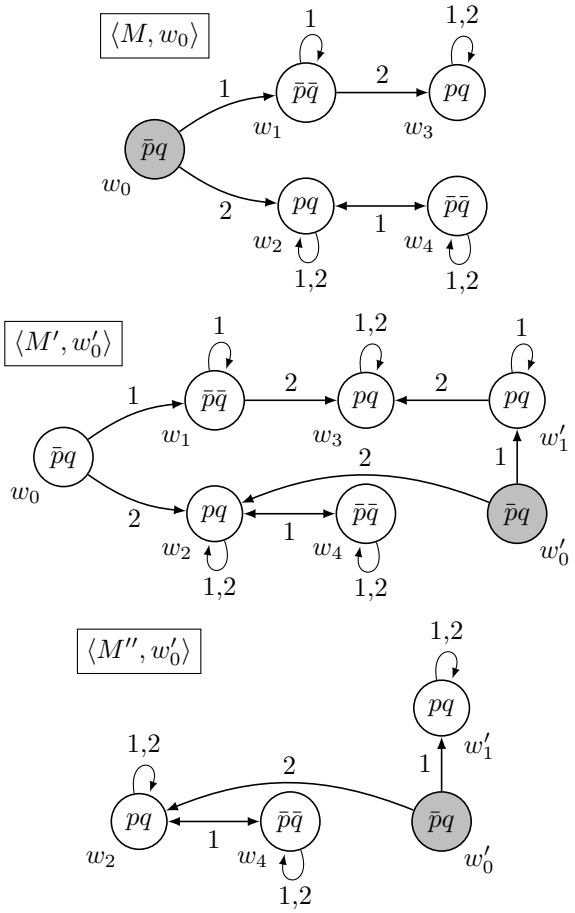


Figure 6.4 – A private revision. On the top, the initial model $\langle M, w_0\rangle$. On the middle, the revised model $\langle M, w_0\rangle \star_1^{*_D} (p \wedge q)$. On the bottom, the smaller model $\langle M'', w_0'\rangle$, which is bisimilar to the model on the middle.

Proposition 6.9. $(\langle M, w_0\rangle \star_i^* \varphi) \leftrightarroweq \langle M.N^{\star_i^*}, w_0.e_0\rangle$.

For example, the event model that corresponds to the revision in Example 6.2 is the following:

$$\begin{aligned}
N^{\star_1} &= \langle E, P, \mathrm{pre}, \mathrm{post}\rangle \\
E &= \{e_0, e_\top, e^{pq}_{w_1}, e^{pq}_{w_2}\} \\
P_1 &= \{(e_0, e^{pq}_{w_1}), (e_0, e^{pq}_{w_2}), (e^{pq}_{w_1}, e^{pq}_{w_1}), (e^{pq}_{w_1}, e^{pq}_{w_2}), (e^{pq}_{w_2}, e^{pq}_{w_2}), (e_\top, e_\top)\} \\
P_2 &= \{(e_0, e_\top), (e_\top, e_\top), (e^{pq}_{w_1}, e_\top)(e^{pq}_{w_2}, e_\top)\} \\
\mathrm{pre}(e_0) &= \neg p \wedge q \\
\mathrm{pre}(e^{pq}_{w_1}) &= p \wedge q \\
\mathrm{pre}(e^{pq}_{w_2}) &= p \wedge q \\
\mathrm{pre}(e_\top) &= \top \\
\mathrm{post}(e^{pq}_{w_1}) &= \{(p, \top), (q, \top)\} \\
\mathrm{post}(e^{pq}_{w_2}) &= \{(p, \top), (q, \top)\}
\end{aligned}$$

The reader may verify that the product between the pointed Kripke model $\langle M, w_0\rangle$ in Figure 6.4 and the model above is bisimilar to the pointed Kripke model $\langle M', w_0'\rangle$ in Figure 6.4.

As for expansion, multiple revision can be achieved with a sequence of private revisions, one for each agent. Common and subjective revision, however, are trickier. Note that the set W^φ in Definition 6.4 is defined using the set $R_a(w_0)$ and an AGM revision of the objective believe set $O_a^{\langle M, w_0\rangle}$. When trying to generalise revision for multiple agents, we then have two problems. The AGM revision operator is only defined for one agent, and it is only defined for objective formulas.

We can think of a way of circumventing this problem. We could replace those sets for their "natural" multi-agent versions, i.e., $R_G(w_0) = \bigcup_{a\in G} R_a(w_0)$ and and $O_G^{\langle M, w_0\rangle} = \bigcap_{a\in G} O_a^{\langle M, w_0\rangle}$. However, it is not completely clear what properties (postulates) a revision operator defined in this way would satisfy. Therefore, this is left as an open question.

6.4 Related Work

There are other studies on the connections between dynamic epistemic logics and belief change theory, e.g., by Baltag and Smets (2006), van Benthem (2007), Board (2004), and van Ditmarsch (2005). These studies investigate how to encode belief change operators within epistemic models with accessibility relations representing plausibility levels, similar to Grove's systems of spheres (Grove 1988). These plausibility levels guide the revision process. Contrastingly, we study here how to perform belief revision (and expansion) on a KD45$_n$ model, representing the beliefs of a group of agents.

In the same vein, Tallon et al. (2004) study what they call revision of KD45$_n$ models due to communication between agents: some agents (publicly) announce (part of) their

beliefs. Their model is closer to expansion than to true revision, and concerns only subjective beliefs.

Herzig et al. (2005) study action progression in multi-agent belief structures. Their work is mainly about the effects of actions using update, but they also briefly mention the problem of revision by objective formulas. Their construction is related to the one we point out, but they do not study the properties of the operators they considered.

Finally, the closest work is the study of private expansion and revision made by Aucher (2008, 2010). The difference is that Aucher considers an internal model of the problem, i.e., a model of the situation viewed from each agent, so he does not use a $\mathrm{KD45}_n$ model for modeling the system, but one internal model by agent. He uses a notion of multi-agent possible worlds in order to compute the result of the revision, so the result of the revision is a set of such multi-agent worlds, whereas in this work we work with $\mathrm{KD45}_n$ models, and we obtain a unique $\mathrm{KD45}_n$ model as result of a revision.

It is easy to find a translation between internal models and $\mathrm{KD45}_n$ models, so one can look at the technical details between the expansion and revision operators we present in this work and the one proposed (on internal models) in (Aucher 2008, 2010). Concerning expansion, it turns out that the two operations are equivalent (that is not surprising since we proved that there is only one rational expansion operator). First, note that it is possible to obtain an internal model I_M for agent $a \in \mathbb{A}$ from any $\mathrm{KD45}_n$ model $\langle M, w\rangle$. Indeed, it suffices to consider the set formed of models $\langle M^k, w^k\rangle$ generated from each w^k such that $w^k \in R_a(w_0)$. Similarly, it is possible to obtain an internal event model $I_{N^{+a}}$ for agent $a \in \mathbb{A}$ from the pointed event model $\langle N^{+a}, e_0\rangle$. Now, it is easy to see that the internal model for a obtained from the product of $\langle M, w_0\rangle$ by $\langle N^{+a}, e_0\rangle$ is the same as the product of I_M by $I_{N^{+a}}$. Concerning revision, the situation is different. Aucher allows revision by subjective formulas and compute distances between the corresponding (epistemic) models. We are interested here only by revision with objective formulas. In this particular case Aucher's revision does not allow the agent concerned by the private revision to choose among the models of the objective formulas, the ones that are the most plausible, this is problematic since this is the basic aim of belief revision. We can do that thanks to the underlying AGM revision operators in the definition of the private revision operator. So our private revision result implies (usually strictly) the result given by Aucher's revision.

6.5 Conclusion

This chapter investigates the problem of belief change in a multi-agent context. More precisely, it studies private expansion and revision of $\mathrm{KD45}_n$ models by objective formulas. We saw a set of postulates for expansion and for revision which are close to the classical AGM ones for the single agent case. We also saw definitions of specific expansion and revision operators that satisfy the desired properties.

All results showed in this chapter would most probably hold for $\mathrm{K45}_n$ models as well. The advantage of using such models is that the absence of axiom (D) permits situations where agents have trivial belief sets, i.e., $K_i^{\langle M,w\rangle} = \mathcal{L}_{\mathrm{ML}}$ (and thus $O_i^{\langle M,w\rangle} = \mathcal{L}_{\mathrm{CPL}}$). Consequently, the assumption that formula φ is consistent can be dropped for both

expansion and revision operators. Similarly, the condition '$\langle M, w\rangle \not\models \mathrm{B}_i\neg\varphi$' can be dropped for postulates (E $+_a$ 1), (E $+_a$ 4). This slightly modified framework is therefore even better as a generalisation of AGM belief change to multi-agent settings.

The problem of revising by subjective formulas is not addressed in this work. As mentioned before, this is more complicated and richer than we studied here due to the minimality of change requirement. A different approach would be to investigate metrics that can be used to define minimal change for revision. In another piece of work not described here (Caridroit et al. 2016), we define and study a number of distances between Kripke models. There, we show how these distances can be used to compute, given an initial model and a formula φ, the "most similar" Kripke model satisfying φ.

Complexity issues of belief change operators in multi-agents settings have not been addressed either. As seen in Section 3.6, KD45$_n$ satisfiability checking is PSPACE-complete. Thus, we have reasons to thrust that belief change in this setting is "more difficult" than in CPL. If that is the case, one way of maybe circumventing this problem is to use fragments of the language. Such kind of approach have been studied, for instance, by Creignou et al. (2016, 2014) and Delgrande and Peppas (2011). Moreover, note that description logics are close related to modal logics, which means that they could also be candidates for settings with better complexities. Approaches for belief change in description logics have been studied, e.g., by Benferhat et al. (2014, 2017).

Some straightforward generalisations to group expansion have been considered in this chapter. Remark, however, that multi-agent belief revision permits even more possibilities than those listed in Section 6.2.3. Recall that the private revision operators $\star$ are defined using AGM revision operators $*$. This means that, in a group revision, each agent could choose a different revision operator. If that were the case, the question now would be: what about the interactions between the agents? I mean, should an agent i assume that another agent j uses the same revision operator as herself? If not, what operator i should chose for j? And, what i thinks that j would chose for i?

It seems that we could model this by adding to the framework yet another epistemic model, where possible worlds would be "possible revision operators", and that new, very general, multi-agent belief revision operators could be defined using this model. However, the way this could be translated into concrete revision operators is left as an open question.

Chapter 7

Methods for Automated Reasoning in Modal Logic

In this chapter, we will study methods for satisfiability checking in some modal logics. This topic is rather different from the ones studied before in this thesis. Nonetheless, I consider this a very important topic for my personal research. The reason is that all the theoretical investigation we saw up till now in this thesis aims at providing formalisms for reasoning. Reasoning about actions, abilities, knowledge, beliefs, responsibility, time, and all this in multi-agent systems. But the idea is not just reason about toy examples and small problems, and certainly not with just pen and paper. The ultimate goal of this research is to provide tools to reason about real life problems, and to do so automatically, with a computer. My intention is to convince people that such a goal is attainable. Therefore, I consider it part of my job to produce some piece of software that performs, at least, some of those reasoning tasks. The contents of this chapter is just the beginning of that endeavour.

This chapter brings together results published with several colleges in a couple of papers, namely, (Caridroit et al. 2017) and (Lagniez et al. 2017). The introduction below motivates the choice of studying logics KT5 and K, and the choice of studying satisfiability checking for these formalisms. Section 7.2 shows the method for satisfiability checking formulas in KT5. It presents the technique used and some experiments showing that such a technique performs well among all other alternatives. Section 7.3 shows a different method of reasoning, this time for satisfiability checking formulas in K. As before, it presents the techniques used and some experiments. Finally, Section 7.4 concludes the chapter.

7.1 Introduction

In the formalisms we saw, the reasoning tasks we want to perform are usually among the following ones:

Consequence checking. For example, assume that, given some conditions described as a set of formulas Γ, we want to know whether agent i would believe that φ under assumptions Γ. In KD45$_n$, it amounts to decide whether, for all pointed Kripke models $\langle M, w\rangle$ satisfying Γ, it is the case that $\varphi \in K_i^{\langle M,w\rangle}$. This is the same as to decide whether $\Gamma \vDash \mathrm{B}_i\varphi$ in KD45$_n$.

Validity checking. For example, assume that we want to know whether the group of agents G can achieve a state of affairs where φ is true, no matter what is the current situation. In ATDEL, it amounts to decide whether $\vDash \langle\!\langle G\rangle\!\rangle\varphi$.

Satisfiability checking. For example, assume that we want to know whether there is a situation where the group G would be responsible for the outcome φ, if they execute joint action δ. In CEDL, it amounts to decide whether there is a model that satisfies $\mathcal{R}_{\delta|_G}\varphi$.

Model checking. For example, assume that we want to know whether the agent i believes that φ is true in a given situation. In KD45$_n$, it amounts to decide whether $\varphi \in K_i^{\langle M,w\rangle}$, which amounts to decide whether $\langle M, w\rangle \vDash \mathrm{B}_i\varphi$.

As saw before (Section 3.6), model checking in modal logics is "easy", in the sense that its computational complexity is polynomial. Consequence, validity and satisfiability checking, though, are another story. First, note that these three tasks are strongly related (see Section 2.1.2). A formula is valid if and only if there is no model for its negation. Therefore, whenever one designs a complete algorithm for satisfiability checking, the same algorithm can be used for validity checking. In the case of consequence, for strongly complete logics, such as KD45$_n$, $\Gamma \vDash \varphi$ is equivalent to $\vDash \bigwedge_{\psi\in\Gamma} \psi \to \varphi$. Therefore, a satisfiability checking algorithm can also be adapted for consequence checking. However, Ladner (1977) showed that the satisfiability problem for several modal logics including K, KT and KT4 are PSPACE-Hard, while it is NP-Complete for single-agent KT5 (see also (Halpern and Rêgo 2007a,b) for more details).

Since recently, SAT solvers (computer programs that solve the satisfiability problem for CPL) have been used as quite efficient NP-oracles for some problems whose complexity is beyond NP (Biere et al. 2009). By the way, SAT solvers have already been used in the context of modal logics (see for example (Sebastiani and Tacchella 2009) for a comprehensive overview of the subject). For instance, *SAT (Giunchiglia and Tacchella 2000b; Giunchiglia et al. 2002) uses a SAT solver to decide satisfiability in 8 different modal logics, including K. A translation of modal logic K to CPL has been proposed in KM2SAT (Sebastiani and Vescovi 2009). More recently, the solver InKreSAT (Kaminski and Tebbi 2013) uses an innovative system, where the SAT solver drives the development of a tableau method.

Curiously, however, none of the methods aforementioned are applicable to modal logic KT5, the easiest one! Single-agent KT5 is considered an "easy" modal logic, because

it is the only one in NP. This means that there is a polynomial algorithm that can translate KT5 to CPL. Therefore, the first thing we do here is exactly that: translate KT5 formulas to CPL and launch a SAT solver. However, we find out that this is less easy than announced.

The second thing we do in this chapter is to attack a more difficult satisfiability checking problem, the one in modal logic K. As for KT5, it is possible to translate K formulas into CPL formulas. But the translated formula can be exponentially larger than the original one. We then find out that the idea of just translating and sending to a SAT solver does not give satisfactory results. The solution described in this chapter is an improvement of the CEGAR (Counter-Example-Guided Abstraction Refinement) approach by Clarke et al. (2003), that has been used in several different problems beyond NP.

7.2 The KT5-SAT problem

Because KT5-SAT, the problem of deciding satisfiability of formulas in KT5, is in NP, we know that there is a polynomial translation from KT5 formulas to CPL formulas. Therefore KT5-SAT is, in principle, "easy": just translate it to CPL and launch the best SAT solver available. However, before our paper (Caridroit et al. 2017), such method has never been tried! We actually did not know if this was better than to use a different method, such as a traditional tableau method for instance. This is so because the translation from KT5 to CPL is not direct. In fact, the translated formula must contain new propositional variables and the CPL formula outputted by the translation is larger than the original formula. Added to the fact that the translation itself might take some time to be performed, this could result in an less efficient method than the traditional tableau methods.

In this section, we try the method of translating KT5 formulas to CPL and launch a SAT solver. But, just to be sure that we are going to win, we propose here a "clever" translation. This translation introduces fewer fresh variables than the standard (naive) translation. We then provide experimental evidence that the proposed approach outperforms the state-of-the-art approaches on the benchmarks considered.

7.2.1 From KT5-SAT to SAT

It has been shown that, if a formula φ with n modal connectives is satisfiable in KT5, then there is KT5-model of φ with at most $n+1$ possible worlds. As a consequence, we know that there exists an algorithm running in polynomial time able to transform the KT5-SAT problem into the SAT problem.

We transform KT5-SAT into SAT with a translation function trkt5. It takes a modal logic formula φ in NNF (Definition 2.15) and a natural number n as input, and produces a formula in $\mathcal{L}_{\text{CPL}}$. This translation is inspired by the standard translation from modal logic to first-order logic (Definition 3.11). Note that the accessibility relation does not need to be represented for a KT5-model, because it is an equivalence relation.

Definition 7.1 (Translation from KT5 to CPL)**.** Let $\varphi \in \mathcal{L}_{\mathrm{ML}}$ be in NNF:

$$
\begin{aligned}
&\mathrm{trkt5}(\varphi, n) = \mathrm{trkt5}(\varphi, 1, n)\\
&\mathrm{trkt5}(p, i, n) = p_i\\
&\mathrm{trkt5}(\neg p, i, n) = \neg p_i\\
&\mathrm{trkt5}(\varphi_1 \wedge \cdots \wedge \varphi_k, i, n) = \mathrm{trkt5}(\varphi, i, n) \wedge \cdots \wedge \mathrm{trkt5}(\varphi_k, i, n)\\
&\mathrm{trkt5}(\varphi_1 \vee \cdots \vee \varphi_k, i, n) = \mathrm{trkt5}(\varphi, i, n) \vee \cdots \vee \mathrm{trkt5}(\varphi_k, i, n)\\
&\mathrm{trkt5}(\Box\varphi, i, n) = \bigwedge_{j=1}^{n} \mathrm{trkt5}(\varphi, j, n)\\
&\mathrm{trkt5}(\Diamond\varphi, i, n) = \bigvee_{j=1}^{n} \mathrm{trkt5}(\varphi, j, n)
\end{aligned}
$$

To compute the translation of φ, it must be in NNF. As mentioned in Section 2.1.4, this operation can be performed in both polynomial time and space. Therefore, without loss of generality, we assume in the remainder of this chapter that the modal logic formulas considered are in NNF.

The translation trkt5 adds fresh propositional variables p_i to the formula. Variable p_i denotes the truth value of p in the possible world w_i. If n is big enough, $\mathrm{trkt5}(\varphi, n)$ is equisatisfiable to φ. This is the case when $n = \mathrm{nm}(\varphi) + 1$, where $nm(\varphi)$ is the number of modal operators in φ.

Proposition 7.1. *$\varphi \in \mathcal{L}_{\mathrm{ML}}$ is satisfiable in* KT5 *if and only if* $\mathrm{trkt5}(\varphi, \mathrm{nm}(\varphi) + 1)$ *is satisfiable in* CPL*.*

The proof of Theorem 7.1 is done in the same way as the standard translation to FOL plus Lemma 6.1 in (Ladner 1977).[1] Also note that the accessibility relation R in KT5 is an equivalence relation. Therefore, it does not need to be represented.

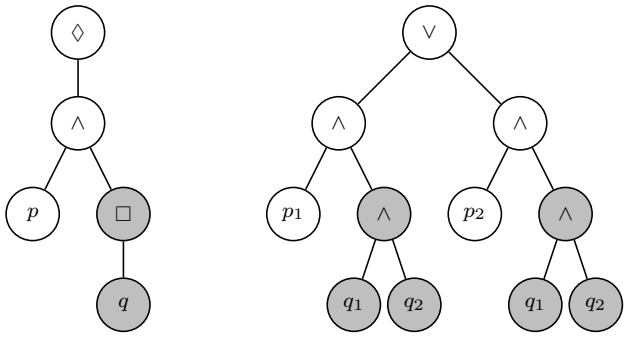
For example, Figure 7.1 shows $\mathrm{trkt5}(\Diamond(p \wedge \Box q), 2)$. Note that the result of the translation is not in CNF. Thus, classical translation into CNF (such as in (Tseitin 1983)) is needed to use a SAT solver. As for NNF, the translation to CNF can also be performed in polynomial time and space.

7.2.2 A New Upper-Bound for the Translation

The length of $\mathrm{trkt5}(\varphi)$ depends on the number of modalities in φ, which we note $\mathrm{nm}(\varphi)$. In practice, this produces unreasonably large formulas. This is why we first tried to find out whether it is possible to decrease the value of n in trkt5. We found a smaller value for it that depends on the *diamond degree* of a formula.

Definition 7.2 (Diamond Degree)**.** The diamond degree of a modal logic formula φ in

[1] The proofs of the results in this section can be found in (Caridroit et al. 2017).



(a) $\Diamond(p \wedge \Box q)$ **(b)** $\mathrm{trkt5}(\Diamond(p \wedge \Box(q))$

Figure 7.1 – Translation from KT5 to CPL

NNF, noted $\mathrm{dd}(\varphi)$, is defined recursively, as follows:

$$\begin{aligned}
&\mathrm{dd}(\top) = \mathrm{dd}(\neg\top) = \mathrm{dd}(p) = \mathrm{dd}(\neg p) = 0\\
&\mathrm{dd}(\varphi \wedge \psi) = \mathrm{dd}(\varphi) + \mathrm{dd}(\psi)\\
&\mathrm{dd}(\varphi \vee \psi) = \max(\mathrm{dd}(\varphi), \mathrm{dd}(\psi))\\
&\mathrm{dd}(\Box\varphi) = \mathrm{dd}(\varphi)\\
&\mathrm{dd}(\Diamond\varphi) = 1 + \mathrm{dd}(\varphi)
\end{aligned}$$

Informally, the diamond degree is an upper bound of the number of diamonds to be taken into account to satisfy the formula.

To show that the diamond degree is a valid value for n, we use a tableau method. Therefore, we need some additional definitions. The tableau defined below is similar to the one in Definition 3.16. The differences are due to the fact that KT5 is simpler than K and, therefore, the tableau can be simplified.

Definition 7.3 (Tableau)**.** Let φ be a KT5 formula in NNF. A tableau for φ is a set of pairs of the form (σ, ψ), where σ is a (possibly empty) sequence of natural numbers and $\psi \in \mathrm{sub}(\varphi)$. In addition, we have that $(0, \varphi) \in T$ and, for all sequences σ, T satisfies the following conditions:

1. $(\sigma, \neg\top) \notin T$.
2. $(\sigma, p) \in T$ if and only if $(\sigma, \neg p) \notin T$.
3. if $(\sigma, \psi_1 \wedge \psi_2) \in T$ then $(\sigma, \psi_1) \in T$ and $(\sigma, \psi_2) \in T$.
4. if $(\sigma, \psi_1 \vee \psi_2) \in T$ then $(\sigma, \psi_1 \in T)$ or $(\sigma, \psi_2) \in T$.
5. if $(\sigma, \Box\psi_1) \in T$ then $\forall(\sigma i, \chi) \in T$ we have $(\sigma, \psi_1) \in T$.
6. if $(\sigma, \Diamond\psi_1) \in T$ then $(\sigma i, \psi_1) \in T$, for some $i \in \mathbb{N}$.

Lemma 7.2. *Let φ be a KT5 formula in NNF. There is a tableau for φ if and only if φ is satisfiable.*

Let T be a tableau for φ. The number of different sequences σ in T depends on the number of times the condition involving operator $\Diamond$ is triggered. It is possible to show, by induction on the structure of φ, that this number is bounded by $\mathrm{dd}(\varphi)$.

Lemma 7.3. *Let φ be a KT5 formula in NNF. The number of different sequences σ in the tableau for φ is bounded by* $\mathrm{dd}(\varphi)$.

Each σi in the tableau T corresponds to a $w_i \in W$ in the KT5-model, $|T| \leq \mathrm{dd}(\varphi)$ means that the number of possible worlds in the model is bounded by $\mathrm{dd}(\varphi)$.

Theorem 7.4. *If $\varphi \in \mathcal{L}_{\mathrm{ML}}$ is satisfiable, then* $\mathrm{trkt5}(\varphi, \mathrm{dd}(\varphi))$ *is satisfiable.*

7.2.3 Structural Caching

Caching is a classical way to avoid redundant work. For instance, the modal logic solver *SAT performs caching using a "bit matrix" (Giunchiglia and Tacchella 2000a). Efficient implementations of BDD packages (such as (Bryant 1986)) also rely on caching to build an explicit graph. These two examples require additional time and space to store and search among already performed work.

The technique we use here is a "simple but efficient" trade-off. It does not memorize the work already done. Thus, it may not cache all possible formulas, but it does not have any additional cost.

As an example, let the formula $\varphi = \Diamond(p \wedge \Diamond q)$ be given. Both translation techniques are depicted in Figure 7.2. The formula in Figure 7.2b contains two copies of $(q_1 \vee q_2)$. This happens because the translation of the first diamond creates two sub-formulas $p_1 \wedge \Diamond q$ and $p_2 \wedge \Diamond q$, where each $\Diamond q$ needs to be translated. Because we are in KT5 (where all possible worlds are connected to each other in the model), both translations of $\Diamond q$ are equivalent. This means that we can reuse the same sub-formula. Therefore, instead of using a tree, we can use a directed acyclic graph (DAG), which also allows us to translate the CPL formula into CNF more efficiently.

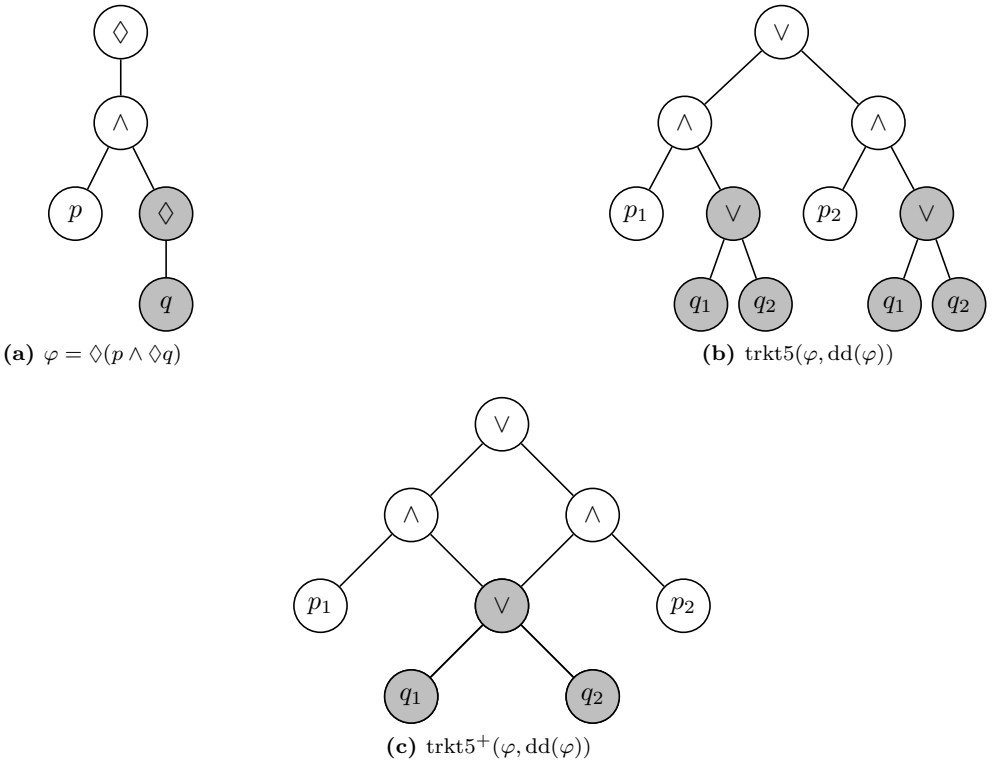
Lemma 7.5. $\mathrm{trkt5}(\circ\varphi, i, n) = \mathrm{trkt5}(\circ\varphi, j, n)\ \forall i, j$ *and* $\circ \in \{\Box, \Diamond\}$

Proof. There are two possible cases:

1. If $(\circ = \Box)$ then, $\mathrm{trkt5}(\Box\varphi, i, n) = \bigwedge_{k=1}^{n} \mathrm{trkt5}(\varphi, k, n) = \mathrm{trkt5}(\Box\varphi, j, n)$
2. If $(\circ = \Diamond)$ then $\mathrm{trkt5}(\Diamond\varphi, i, n) = \bigvee_{k=1}^{n} \mathrm{trkt5}(\varphi, k, n) = \mathrm{trkt5}(\Diamond\varphi, j, n)$

Therefore, we have $\mathrm{trkt5}(\circ\varphi, i, n) = \mathrm{trkt5}(\circ\varphi, j, n)$. ∎

Informally, Lemma 7.5 implies that no matter how deep in the tree the sub-formula $\circ\varphi$ is, its translation is always the same. Therefore, we can translate the deepest sub-formula first and then backtrack. For example, in Figure 7.2(b), the same sub-formula appears twice, which means that we can use only one of its occurrences, such as in Figure 7.2(c). We denote this modified translation function $\mathrm{trkt5}^{+}$. Structural caching is thus performed on the fly when translating the modal logic formula into propositional logic, before translating the formula in CNF.



(a) $\varphi = \Diamond(p \wedge \Diamond q)$

(b) $\mathrm{trkt5}(\varphi, \mathrm{dd}(\varphi))$

(c) $\mathrm{trkt5}^{+}(\varphi, \mathrm{dd}(\varphi))$

Figure 7.2 – Comparison between trkt5 and trkt5^{+}

Solver	solved	SAT	MO	TO
LckS5TabProver	709	143	710	655
S52SAT nm	1377	411	667	30
S52SAT nm+	1733	452	292	49
S52SAT dd	1645	433	412	17
S52SAT dd+	**1834**	**460**	**203**	37
SPASS 3.7	1530	451	528	**16**

Table 7.1 – Overall results. solved: number of solved instances, SAT: number of satisfiable instances, MO: number of memory outs, TO: number of time outs.

7.2.4 Experiments

Several experiments were performed to evaluate the techniques aforementioned using the solver S52SAT, implemented by Valentin Montmirail. We will not see all the results here. The interested reader may find more information in (Caridroit et al. 2017) and in Montmirail's thesis (Montmirail 2018).

Table 7.1 summarises the results obtained in all benchmarks. The best result for each column is in bold face. At the time we published our results, LckS5TabProver, developed by Abate et al. (2007) and SPASS, developed by Weidenbach et al. (2009), were the state-of-the art in S5-satisfiability checking. Here, we compare them with 4 different versions of S52SAT:

- S52SAT nm uses the number of modalities of the formula as the upper-bound for the translation;
- S52SAT nm+ does the same but also uses the caching technique;
- S52SAT dd uses the diamond degree as the upper bound;
- S52SAT dd+ uses the diamond degree and the caching.

In all versions, Glucose 4.0 (Audemard and Simon 2009) is used as the internal CPL SAT solver.

The evaluations were performed using well known modal logic benchmarks: 3CNFK (Patel-Schneider and Sebastiani 2003), MQBFK (Massacci 1999), T ANCS 2000K (Massacci and Donini 2000) and LWB K, KT, S4 Balsiger et al. (2000). Note that they are designed for modal logics K, KT and S4. As a consequence, some of them are trivial in KT5. However, the results are still significant, since KT5 contains K, KT and KT4 (see Figure 3.2).

7.3 The K-SAT problem

To attack the K-SAT problem, we need more than just a clever translation. This is so because, no matter how clever it may be, the output of the translation can be exponentially larger than the original formula. This means (and sometimes is the case in

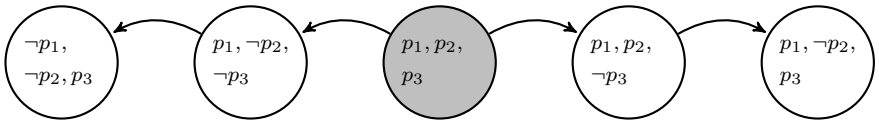
practice) that the translated formula may not fit into the computer memory. When that happens, it is just not possible to find a model for the formula.

To make things even more difficult, the diamond degree cannot be used in modal logic K. The example below shows us why.

Example 7.1. Let the formula φ be:

$$
\begin{aligned}
&(p_1 \wedge p_2 \wedge p_3) \wedge \\
&(\Diamond(p_1 \wedge p_2 \wedge \neg p_3 \wedge \Box(p_1 \wedge \neg p_2 \wedge p_3))) \wedge \\
&(\Diamond(p_1 \wedge \neg p_2 \wedge \neg p_3 \wedge \Box(\neg p_1 \wedge \neg p_2 \wedge p_3))) \wedge \\
&(\Box\Diamond p_3)
\end{aligned}
$$

Its diamond degree is $\mathrm{dd}(\varphi) = 3$. This formula is satisfied, e.g., by the model below.



However, the reader may verify that it is not possible to find a model satisfying φ with less than 5 possible worlds. ■

To handle the K-SAT problem, we use here an improvement of the CEGAR approach (Clarke et al. 2003), whose the original idea is as follows. Imagine a very large CPL formula in CNF. Instead of giving it directly to the SAT solver, we give only part of it, and then ask for a model. If the SAT solver answers UNSAT, then the entire formula is UNSAT. Our program can just return UNSAT and we are done (this is the UNSAT shortcut). On the other hand, if the SAT solver gives us a model, then it is easy (can be done in linear time) to verify if that model is a model for the entire formula. If we are lucky, it is so. In this case, the program can just return that model and we are done. If we are not so lucky (i.e., the SAT solver find a model that does not satisfy the entire formula) then we go for another run: we add some constraints to prevent the sat solver from finding the same model again and give it a bigger part of the original formula, hoping to be luckier this time (this is the refinement step). In the worst case, either the program runs out of memory (in which case the program returns UNKNOWN), or we end up giving the entire original formula to the SAT solver (in which case it will decide its satisfiability anyway).

This framework has been applied in various areas: Satisfiability Modulo Theory (Brummayer and Biere 2009), Planning (Seipp and Helmert 2013) and, more recently, QBF (Janota et al. 2016). Several previous SAT-based approaches have already been proposed in the field of modal logic (Sebastiani and Vescovi 2009). One could argue that *SAT (Giunchiglia et al. 2002) is already a CEGAR approach for the modal logic K.

In this chapter, we present an extension of CEGAR which includes a recursive step to introduce a new shortcut in the original procedure. We call this extension *Recursive Explore and Check Abstraction Refinement (RECAR)*. The idea of mixing SAT and UNSAT shortcuts in a CEGAR procedure is not new: it has been already used for SMT

(Brummayer and Biere 2009) and for bug detection (Wang et al. 2007). The novelty here is that we use an abstraction of the original problem in the loop, made possible by a recursive call to the main procedure.

The RECAR procedure is generic, i.e., it is not bound to a specific domain. We present the conditions required on the abstractions used, and the correctness of the approach. We instantiate the framework for the satisfiability of modal logic K, by providing abstraction functions for this problem and experimental results of the approach against state-of-the-art solvers.

7.3.1 CEGAR Preliminaries

Counter-Example-Guided Abstraction Refinement (CEGAR) is an incremental way to decide the satisfiability of formulas in CPL. It has been originally designed for model checking (Clarke et al. 2003), i.e., to answer questions such as 'does $\Gamma \vDash \psi$?' or, equivalently, "is $\Gamma \wedge \neg\psi$ unsatisfiable?", where Γ describes a system and ψ a property. In such highly structured problems, it is often the case that only a small part of the formula is needed to answer the question. The idea behind CEGAR is to replace $\varphi = \Gamma \wedge \neg\psi$ by an approximation φ', where φ' is easier to solve in practice than φ. There are two kinds of approximations:

- an over-approximation of φ is a formula $\hat{\varphi}$ such that $\hat{\varphi} \vDash \varphi$ holds, i.e., $\hat{\varphi}$ has at most as many models as φ;
- an under-approximation of φ is a formula $\check{\varphi}$ such that $\varphi \vDash \check{\varphi}$ holds, i.e., $\check{\varphi}$ has at least as many models as φ.

Usually, φ is in CNF. Then, a classical way to under-approximate φ is to remove some clauses from it. In other words, $\check{\varphi}$ is a subset of the clauses in φ. A model of $\check{\varphi}$ also may by chance satisfy φ. Moreover, if $\check{\varphi}$ is found to be unsatisfiable, then so is φ. This double possibility to conclude earlier makes under-approximation based CEGAR very popular. A classical way to over-approximate is to bound the generation of the formula φ to a given number n smaller than the one needed to reach equisatisfiability to the original problem (as in bounded model checking (Clarke et al. 2003) or planning (Seipp and Helmert 2013)). In this case, a model of $\hat{\varphi}$ can be extended to a model of φ, but the unsatisfiability of φ means that n has to be incremented and the process repeated.

An example of a CEGAR algorithm using over-approximations is given on Algorithm 7.1. It receives a formula φ as input and computes an over-approximation ψ. Then it uses a oracle (usually a SAT solver) to check whether ψ is satisfiable. If so, it returns SAT. Otherwise, ψ is refined (i.e., it gets closer to φ) until it is satisfiable, or until the refined over-approximation is detected to be equisatisfiable to φ. Function eqsat(φ, ψ) denotes an incomplete but efficient equisatisfiability test, which returns **true** if it is able to detect that $\varphi \overset{\text{sat}}{\equiv} \psi$ and **false** otherwise.

Recent SAT solvers are able to check *satisfiability under assumptions* (Eén and Sörensson 2003). This means that, assuming a given set of literals are satisfied, provide, in case of unsatisfiability, a "reason", in terms of those literals, for the unsatisfiability of a formula. Formally, we have the following.

```
1 function cegar(φ)
2     ψ ← over(φ)
3     while not check(ψ) do
4         if eqsat(ψ, φ) then
5             return UNSAT
6         ψ ← refine(ψ)
7     return SAT
```

Algorithm 7.1: CEGAR with over-approximation

Definition 7.4 (Unsatisfiable Core with Assumptions)**.** Let φ be in CNF and A be a consistent set of literals from φ. Let φ be satisfiable and $(\varphi \wedge \bigwedge_{a \in A} a)$ be unsatisfiable. $L \subseteq A$ is an unsatisfiable core of φ if and only if $(\varphi \wedge \bigwedge_{l \in L} l)$ is unsatisfiable.

Therefore, a SAT solver for φ, given A, can be seen as a procedure providing a pair (d, r) with $d \in \{\text{SAT}, \text{UNSAT}\}$ and r is a model of φ if $d = \text{SAT}$ or an unsatisfiable core of φ if $d = \text{UNSAT}$. Modern SAT-based procedures are able to take r into account in both cases. Unsatisfiable cores have been used, for instance, in a CEGAR approach for deciding satisfiability of formulas in the propositional fragment of first-order logic (Khasidashvili et al. 2015).

7.3.2 Recursive Explore and Check Abstraction Refinement

A classic CEGAR approach with over-approximation and a SAT shortcut performs well when the input is satisfiable. But, generally, it does not perform well when the input is unsatisfiable. The reason is that it may have to keep refining until it reaches equisatisfiability with the original problem. One way to address this issue is to mix SAT and UNSAT shortcuts, as in (Brummayer and Biere 2009) and (Wang et al. 2007). In these approaches, the methods alternate between over and under approximations.

Recursive Explore and Check Abstraction Refinement (RECAR), depicted in Algorithm 7.2, interleaves both kinds of approximations. Each abstraction is performed with the information retrieved from solving the previous one. The UNSAT shortcut is implemented using a recursive call to the main procedure when a strict under-approximation $\breve{\varphi}$ can be built. It is function 'rc()' that verifies if the under-approximation is strict.

One should also note that the proposed approach permits abstractions on two different levels: one is used to simplify the problem at the domain level (recursive call), while the other one is used to approximate the problem at the oracle (function 'check()') level.

In order to be able to apply RECAR, the under- and over-approximations must satisfy some properties. In the following, we use $\models_1$ and $\models_2$ to denote two possibly different consequence relations (two different logics). Let $\text{under}(\varphi) = \breve{\varphi}$. We also use $\text{refine}^n()$ and $\text{under}^n()$ to indicate n successive applications of functions refine() and under(), respectively.

```
1  function recar(φ)
2      ψ ← over(φ)
3      while true do
4          if check(ψ) = SAT then
5              return SAT
6          if eqsat(ψ, φ) then
7              return UNSAT
8          χ ← under(φ)
9          if rc(φ, χ) then
10             if recar(χ) = UNSAT then
11                 return UNSAT
12         ψ ← refine(ψ)
```

Algorithm 7.2: Recursive Explore and Check Abstraction Refinement

RECAR Assumptions.

1. Function 'check()' is a sound and complete implementation of '$\models_1$' which terminates.
2. If $\not\models_1 \neg\text{over}(\varphi)$ then $\not\models_1 \neg\text{refine}(\text{over}(\varphi))$.
3. $\text{eqsat}(\text{refine}^n(\hat{\varphi}), \varphi)$ returns **true**, for some $n \in \mathbb{N}$.
4. If $\models_2 \neg\check{\varphi}$ then $\models_2 \neg\varphi$.
5. $\text{rc}(\text{under}^n(\varphi),\text{under}^{n+1}(\varphi))$ returns **false**, for some $n \in \mathbb{N}$.

Note that, if $\hat{\varphi}$ is satisfiable, then φ is satisfiable, by assumptions 2 and 3. In the following, we show that, under these assumptions, RECAR is sound, complete and terminates.

Proposition 7.6 (Soundness)**.** *If* $\text{recar}(\varphi)$ *returns SAT then* φ *is satisfiable.*

Proof. Assume that $\text{recar}(\varphi)$ returns SAT. This is the case only if $\text{check}(\psi)$ returns SAT, on line 4 of Algorithm 7.2. Thus, we know that ψ is satisfiable (by Assumption 1). But ψ equals to $\text{over}(\varphi)$ or equals to $\text{refine}^n(\varphi)$, for some $n \in \mathbb{N}$. Then φ is satisfiable (by assumpumptions 2 and 3). ∎

The intuition behind the proof of Theroem 7.7[2]below is that there are two ways to conclude that φ is unsatisfiable. In the first case, $\hat{\varphi}$ is refined a finite number of times until it is detected equisatisfiable to φ, and check() returns UNSAT. Then φ is unsatisfiable. In the second case, one of the under-approximations is shown to be unsatisfiable, then φ is unsatisfiable, by Assumption 4).

[2]Proofs can be fond in (Lagniez et al. 2018).

Theorem 7.7 (Completeness)**.** *If* recar(φ) *returns UNSAT then* φ *is unsatisfiable.*

The intuition behind the proof of Theorem 7.8 below is that the function performs a finite number of recursive calls (by Assumption 5). Moreover, each of these calls has a finite number of refinements before terminating (by Assumption 3).

Theorem 7.8 (Termination)**.** *RECAR terminates for any input* φ*.*

7.3.3 An Implementation of RECAR for Modal Logic

In order to be able to apply RECAR to modal logic K, we need to find over- and under-approximations respecting all five RECAR assumptions. First, we show the over-approximation used in our solution.

Over-approximation

The over-approximation function used in our K-SAT solver uses a translation from ML to CPL that is similar to the one in Definition 7.1. As already discussed, in the general case, this translation outputs a formula that is exponentially larger than the original one. The idea though, is to avoid using the translation naively. But first, let us see its definition. We assume a formula φ in NNF.

Definition 7.5 (Translation from K to CPL)**.**

$$
\begin{aligned}
\mathrm{mltr}(\varphi, n) &= \mathrm{mltr}(\varphi, 0, n) \\
\mathrm{mltr}(p, i, n) &= p_i \\
\mathrm{mltr}(\neg p, i, n) &= \neg p_i \\
\mathrm{mltr}(\varphi \wedge \psi, i, n) &= \mathrm{mltr}(\varphi, i, n) \wedge \mathrm{mltr}(\psi, i, n) \\
\mathrm{mltr}(\varphi \vee \psi, i, n) &= \mathrm{mltr}(\varphi, i, n) \vee \mathrm{mltr}(\psi, i, n) \\
\mathrm{mltr}([m]\varphi, i, n) &= \bigwedge_{j=0}^{n} (\neg r^a_{i,j} \vee \mathrm{mltr}(\varphi, j, n)) \\
\mathrm{mltr}(\langle m \rangle \varphi, i, n) &= \bigvee_{j=0}^{n} (\neg r^a_{i,j} \vee \mathrm{mltr}(\varphi, j, n))
\end{aligned}
$$

Let φ be the input formula. As before, we have that, if $\mathrm{mltr}(\varphi, \mathrm{nm}(\varphi))$ is satisfiable in CPL then φ is satisfiable in K. However, $\mathrm{mltr}(\varphi, \mathrm{nm}(\varphi))$ can be very large. We therefore start with $\mathrm{mltr}(\varphi, n)$, for some small number n. If we find a model for this formula, then φ has a model, we can stop and return SAT. If we do not find a model, we increment n and start again. Formally, we have.

Definition 7.6 (Over-approximation)**.** Let $\varphi \in \mathcal{L}_{\mathrm{ML}}$. The over-approximation of φ, denoted $\hat{\varphi}$, is the formula $\mathrm{mltr}(\varphi, 1)$.

Definition 7.7 (Over-approximation Refinement)**.** Let $1 \leq n \leq \mathrm{nm}(\varphi)+1$. The refinement of $\mathrm{mltr}(\varphi, n)$, noted $\mathrm{refine}(\mathrm{mltr}(\varphi, n))$ is the formula $\mathrm{mltr}(\varphi, n+1)$.

Theorem 7.9. *If* $\mathrm{mltr}(\varphi, n)$ *is satisfiable then* $\mathrm{mltr}(\varphi, n+1)$ *is satisfiable, for all* $1 < n \leq \mathrm{nm}(\varphi)+1$.

Proof Sketch. The idea is that if φ is satisfied by a model M with n worlds, then we can find a model M' with $n+1$ worlds satisfying φ. The additional world is just not accessible from the ones already in M. ∎

Theorem 7.9 above means that RECAR Assumption 2 is satisfied by this over-approximation. Assumption 3 is also satisfied, because $\mathrm{mltr}(\varphi, n) \overset{\text{sat}}{\equiv} \mathrm{mltr}(\varphi, n+1)$, for $n > \mathrm{nm}(\varphi)$. This allows us to use this over-approximation and refinement in the RECAR approach.

Under-approximation

To understand the intuition behind the under-approximation we use in our solver, let us see an example. Let $\varphi = (\Diamond p \wedge \Box\neg p \wedge \chi)$ for some $\chi \in \mathcal{L}$, where $\mathrm{nm}(\chi)$ is huge. This is clearly unsatisfiable because $(\Diamond p \wedge \Box\neg p)$ is unsatisfiable. One can see that right away without even knowing what χ looks like. However, a CEGAR approach using the over-approximation and refinement defined earlier will take a long time before finally conclude it. The reason is that each refinement $\mathrm{mltr}(\varphi, n+1)$ of the original formula will be shown unsatisfiable and it will not stop until the huge number $\mathrm{nm}(\varphi)+1$ is reached.

We found a way to avoid these pathological cases, as follows. Let us take that formula φ again. First, we add to each conjunct in φ a fresh variable s_i (a selector) that will be assumed to be true by the SAT solver, as done in Figure 7.3. Then, we make the first over-approximation $\mathrm{mltr}(\varphi, 1)$ and give it to a modern SAT solver. The solver will return UNSAT with an unsatisfiable core. From this core, we extract a set of selectors *core*. Let us assume, in our example, that $core = \{s_1, s_2\}$. This means that the formula $\check{\varphi} = (\Diamond p \wedge \Box\neg p)$, which is the one labelled by the selectors, is enough to prove the unsatisfiability of φ with only 1 possible world. Proving the unsatisfiability of $\check{\varphi}$ will imply that φ is unsatisfiable. Note that, in this specific case, $\mathrm{nm}(\check{\varphi})$ is much smaller than $\mathrm{nm}(\varphi)$. Thus, if we launch a SAT solver on $\mathrm{mltr}(\check{\varphi}, 1)$ will succeed right a way, while it may have failed for the entire formula φ. Formally, we have the following.

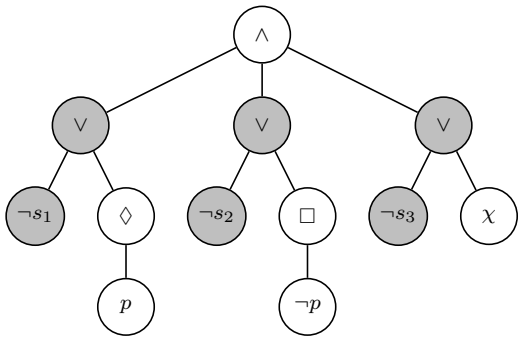


Figure 7.3 – The formula $\varphi = (\Diamond p \wedge \Box \neg p \wedge \chi)$ with selectors

Definition 7.8 (Under-approximation)**.**

$$\begin{aligned}
&\text{under}(p, core) = p \\
&\text{under}(\neg p, core) = \neg p \\
&\text{under}([m]\varphi, core) = [m](\text{under}(\varphi, core)) \\
&\text{under}(\langle m\rangle\varphi, core) = \langle m\rangle(\text{under}(\varphi, core)) \\
&\text{under}((\varphi \wedge \psi), core) = \text{under}(\varphi, core) \wedge \text{under}(\psi, core) \\
&\text{under}((\psi \vee \chi), core) = \begin{cases} \text{under}(\chi, core) & \text{if } \psi = \neg s_i, s_i \in core \\ \top & \text{if } \psi = \neg s_i, s_i \notin core \\ (\text{under}(\psi, core) & \\ \vee \text{under}(\chi, core)) & \text{otherwise} \end{cases}
\end{aligned}$$

Theorem 7.10. *If* $\text{under}(\varphi, core)$ *is unsatisfiable then* φ *is unsatisfiable.*

The intuition of the proof is that each selector s_i enables an operand in a conjunction of the formula. Each time function under() is called with a non-empty *core*, operands not enabled with a selector from the *core* will be removed from the formula.

Theorem 7.10 shows that function under() satisfies RECAR Assumption 4. To see that it also satisfies Assumption 5, note that the length of $\text{under}^{n+1}(\varphi, core)$ is smaller or equal to that of $\text{under}^{n}(\varphi, core')$ (even though the sets *core* and $core'$ usually differ).

MoSaiC is RECAR for K-SAT

Algorithm 7.3 shows the instantiation of RECAR to modal logic K satisfiability problem. It uses the over- and under-approximation defined in the previous sections.

This algorithm has been implemented by Valentin Montmirail in the solver MoSaiC. As before, MoSaiC uses the SAT solver Glucose (Audemard et al. 2013; Eén and Sörensson 2003) to decide the satisfiability of each ψ.

Some implementation details are not depicted in the algorithm. For instance, Glucose is not called on ψ but on an updated ψ' with selectors on conjuncts under the assumption

```
1  function mosaic(φ)
2      l ← 1
3      ψ ← mltr(φ, l)
4      while true do
5          if glucose(ψ) then
6              return SAT
7          if l > nm(φ) then
8              return UNSAT
9          χ ← under(φ, core)
10         if χ ≠ φ then
11             if check(χ) = UNSAT then
12                 return UNSAT
13             else
14                 l ← max(|M|, l + 1)
15         ψ ← mltr(φ, l)
```

Algorithm 7.3: MoSaiC

that these selectors are satisfied. It is not necessary to generate the under approximation χ to test the condition $\chi \neq \varphi$ on line 10. It suffices to know the number of selectors involved in the unsatisfiability of the formula. MoSaiC also returns a Kripke model, when it finds one. This information is used to calculate the new value of l. Finally, note that, in this specific case, $\max(|M|, l+1)$ always returns $|M|$, because it is not possible to find a model smaller than M by the definition of under(φ).

7.3.4 Experiments

As before, only some results are shown here. The reader can find more information in (Lagniez et al. 2017) and in Montmirail's thesis (Montmirail 2018).

MoSaiC has been compared against the same solvers as in (Nalon et al. 2016), namely:

- K_SP 0.1 (Nalon et al. 2016);
- BDDTab 1.0 (Goré et al. 2014);
- FaCT++ 1.6.4 (Tsarkov and Horrocks 2006);
- InKreSAT 1.0 (Kaminski and Tebbi 2013);
- *SAT 1.3 (Giunchiglia et al. 2002);
- Km2SAT 1.0 (Sebastiani and Vescovi 2009) combined with the same Glucose SAT solver we use in MoSaiC;
- Spartacus 1.0 (Götzmann et al. 2010);

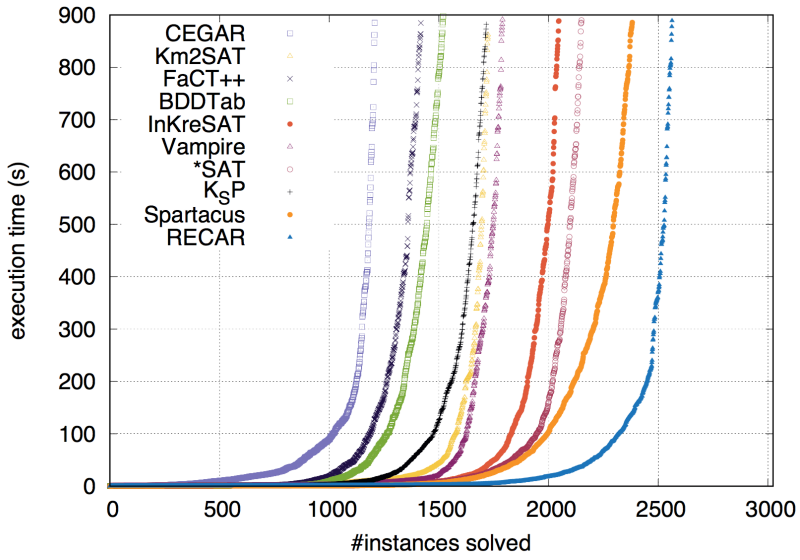


Figure 7.4 – Runtime distribution on all the benchmarks

- a combination of the optimized functional translation (Horrocks et al. 2007) with Vampire 4.0 (Kovács and Voronkov 2013).

We can see on Figure 7.4 that MoSaiC with CEGAR is the worst solver whereas MoSaiC with RECAR outperforms all other solvers. Km2SAT has a specific technique to detect some unsatisfiable formulas without generating the CNF. This explains why it performs much better than the CEGAR approach. *SAT interleaves SAT reasoning and domain reasoning, and can be considered as an under-approximation CEGAR approach. It shows good results, despite being tied with the old SAT solver SATO. Spartacus is based on a tableau method, not on SAT. SAT based-techniques were not the best way to tackle such problems up to this work.

7.4 Conclusion

We saw in this chapter a reduction from KT5 to CPL that has been used to address the KT5-SAT problem using a SAT solver. This reduction uses a lower upper bound on the number of possible worlds of the model and also structural caching. We saw a comparison of this approach against solvers representing the state-of-the-art for KT5-SAT in a number benchmarks. The approach presented here outperforms those solvers.

Even if the benchmarks were not KT5 benchmarks, since they come from other modal logics, namely, K, KT and KT4, those results open interesting perspectives. For instance, satisfiability in KT5 entails satisfiability on less restrictive models, such as K, KT and KT4. Since S52SAT finds models in just a few seconds (2.06s median time), it could eventually be used as a preprocessing step for other modal logics.

The second part of this chapter presented the algorithm called recursive explore and check abstraction refinement (RECAR). We saw that, if the five RECAR conditions are fulfilled, the underlining algorithm is sound, complete and terminates. RECAR has been instantiated for the K-SAT problem in the solver MoSaiC, which has been compared to several other solvers on a number of benchmarks. It outperformed all those solvers on the benchmarks considered.

The natural continuation of this research is tackling the remaining modal logics in Figure 3.2. Actually, we have already started it. In a recent publication (Lagniez et al. 2018), we showed that, with some adaptations, MoSaiC can address all modal logics with axioms (T), (B), (4) and (5). The first experiments already show that MoSaiC is competitive on logics KT and KT4, but more improvements are on the way.

Chapter 8

Conclusion

This document describes part of the research I did in the past eleven years. Here, we saw that, in the first years, I worked on the formalisation of individual and collective responsibility in order to formalise the problem of many hands. To attain that objective, I designed coalition epistemic dynamic logic, a formalism capable of expressing agents actions, knowledge and, through abbreviations, also ability, knowing ability and obligation. The logic designed was apt to the task. Formalisations of different kinds of responsibility and also of the problem of many hands were proposed using this logic.

Towards the end of that first period, in part motivated by a number of questions left open for CEDL, I decided to design a similar, but more suitable formalism. The alternating-time dynamic epistemic logic has actions in "dynamic epistemic style", which permitted to address those open questions, as well as reasonable sized systems specifications.

Some years passed and the next natural step was enriching ATDEL. There were several possibilities. I decided that an interesting development would be the possibility of modeling environments where agents can have incorrect information, and where they are able to reconsider their knowledge base when they realise their mistakes. This lead me to the somewhat different area of belief revision theory. I then worked on that area with colleagues at the CRIL, aiming at finding generalisations of expansion and revision for multi-agents scenarios.

That topic is not yet closed. Some interesting possibilities for advancement of the work on belief revision are still possible. As it has been pointed out before, it comprises multi-agent revision by subjective formulas, and a very general kind of belief revision where different agents use different ways to revise their beliefs.

More recently, I decided to search for ways of putting in practice some of the theoretical investigations I did all these years. This lead me to the study of modal logic automated reasoning. Indeed, in the last part of this document, I described some research carried out with CRIL colleagues on that subject. Putting together our different expertises, the system built from that collaboration were able to outperform state-of-

the-art techniques on modal logic automated reasoning.

This thesis obviously does not describe all the research work I did in the period. One interesting result not described here is (Herzig et al. 2009). On that paper, we investigate a logic called acceptance logic, which aims at modelling individual and collective acceptances. In that logic, formulas of the form $A_{G:x}\varphi$ read 'if the agents in the group G identify themselves with institution x then they together accept that φ'.

We extend that formalism by two dynamic modal operators. The first one models the event of agents learning some piece of information ψ in some context x. Technically, this is done using public announcements, i.e., public actions that make all agents in the environment commonly know that ψ is true. In the case here, ψ becomes commonly accepted in context x. The second operator models the event of agents shifting (changing) their acceptances in order to accept some piece of information ψ in context x.

We propose a sound and complete axiom system for this dynamic extension of acceptance logic. In addition, we show that this formalism can be used to model some interesting aspects of judgement aggregation. In particular, we apply it to the 'doctrinal paradox' (Kornhauser and Sager 1986; Petit 2001). Because of a distinction between what agents accept in a context x and what agents believe, the doctrinal paradox does not necessarily lead to a contradiction, when formalised in that logic.

Another interesting result in the period is (Magnier and de Lima 2015). On that paper, we create a dialogical version of public announcement logic. In this version, the validity of a formula is defined via an argumentative game between two adversaries, the proponet and the opponent. The proponent starts a game by uttering a thesis (a formula) and then tries to defend it against the opponent. The opponent tries to refute the proponent's thesis. The moves in this game are either challenges on adversary's moves or defenses against adversary's challenges. The game is played with players acting alternatively. The game terminates when there are no more allowed moves. The proponent wins if there are no allowed moves for the opponent, and vice-versa.

We showed soundness and completeness of the game rules. The rules are made in such a way that simple adaptations can handle other systems, e.g., systems without the truth axiom, positive or negative introspection.

Future plans include the natural continuations aforementioned and a couple of new ideas.

For instance, I have started co-supervising a third Ph.D., now on the subject of intentions in epistemic games. The latter are games where players must have to take other players knowledge or beliefs into account to be able to play effectively. Common examples are Cluedo, Hanabi and most card games.

In this kind of game, some moves may reveal player's intentions. For example, in a collaborative game like Hanabi, a player may try to communicate with her partner by playing some specific card. In a competitive game, a player may try to make the opponent believe that she has some important card, in order to make the opponent take a "bad" decision.

The project has two phases. The first one is to propose a logical modelling of intentions in epistemic games. This can be done using the formalisms we saw here, i.e., logics of action, knowledge, belief, abilities, etc.. The second phase is to use this modelling to explain players decisions in the course of games.

Detection of players intentions is generally computationally expensive. The time allotted to each player is often too short to allow performing this task during the game. But it is possible to study games off-line in order to, e.g., identify bad moves or explain good ones. This second phase will use the results on modal logic automated reasoning.

Appendix A

List of Publications by T. de Lima

Articles in International Journals

Balbiani, Philippe, Alexandru Baltag, Hans van Ditmarsch, Andreas Herzig, Tomohiro Hoshi, and Tiago de Lima (Oct. 2008). "'Knowable' as 'known after an announcement'". In: *The Review of Symbolic Logic* 1.3, pp. 305–334. DOI: 10.1017/S1755020308080210.

Balbiani, Philippe, Hans van Ditmarsch, Andreas Herzig, and Tiago de Lima (2010). "Tableaux for Public Announcement Logic". In: *Journal of Logic and Computation* 20.1, pp. 55–76. DOI: 10.1093/logcom/exn060.

van Ditmarsch, Hans, Andreas Herzig, and Tiago de Lima (2011). "From Situation Calculus to Dynamic Epistemic Logic". In: *Journal of Logic and Computation* 21.2, pp. 179–204. DOI: 10.1093/logcom/exq024.

— (2012). "Public announcements, public assignments and the complexity of their logic". In: *Journal of Applied Non-Classical Logics* 22.3, pp. 249–273. DOI: 10.1080/11663081.2012.705964.

Herzig, Andreas, Tiago de Lima, and Emiliano Lorini (2009b). "On the dynamics of institutional agreements". In: *Synthese* 171, pp. 321–355. DOI: 10.1007/s11229-009-9645-2.

de Lima, Tiago (2014). "Alternating-time temporal dynamic epistemic logic". In: *Journal of Logic and Computation* 24.6, pp. 1145–1178. DOI: 10.1093/logcom/exs061.

de Lima, Tiago, Lambèr Royakkers, and Frank Dignum (2010b). "A Logic for Reasoning about Responsibility". In: *Logic Journal of the IGPL* 18.1, pp. 99–117. DOI: 10.1093/jigpal/jzp073.

Magnier, Sébastien and Tiago de Lima (June 2015). "A soundness & completeness proof on dialogs and dynamic epistemic logic". In: *Logique & Analyse* 230, pp. 219–250. DOI: 10.2143/LEA.230.0.3141809.

Articles in International Conference and Workshop Proceedings

Balbiani, Philippe, Alexandru Baltag, Hans van Ditmarsch, Andreas Herzig, Tomohiro Hoshi, and Tiago de Lima (2007b). "What can we achieve by arbitrary announcements?: A Dynamic Take on Fitch's Knowability". In: *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2007)*. Ed. by D. Samet. ISBN: 978-2-87463-077-4. Presses Universtaires de Louvain, pp. 42–51. ISBN: 978-2-87463-077-4.

Balbiani, Philippe, Hans van Ditmarsch, Andreas Herzig, and Tiago de Lima (2007). "A Tableau Method for Public Announcement Logics". In: *Automated Reasoning with Analytic Tableaux and Related Methods, 16th International Conference, (TABLEAUX 2007)*. Ed. by Nicola Olivetti. Vol. 4548. Lecture Notes in Computer Science. Springer, pp. 43–59.

— (2012). "Some truths are best left unsaid". In: *Advances in Modal Logic Volume 9*. Ed. by Thomas Bolander, Torben Braüner, Silvio Ghilardi, and Lawrence S. Moss. College Publications, pp. 36–54.

de Boer, Mathijs, Andreas Herzig, Tiago de Lima, and Emiliano Lorini (2009). "Tableaux for Acceptance Logic". In: *Declarative Agent Languages and Technologies VII*. Ed. by Mateo Baldoni, Jamal Bentahar, M. Birna van Riemsdijk, and John Lloyd. Vol. 5948. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-11355-0_6. Springer, pp. 85–100.

Caridroit, Thomas, Sébastien Konieczny, Tiago de Lima, and Pierre Marquis (2015a). "Private Expansion and Revision in Multi-Agent Settings". In: *Proceedings of the 13th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2015)*. Ed. by Sébastien Destercke and Thierry Denoeux. Vol. 9161. Lecture Notes in Computer Science. Springer, pp. 175–185.

— (2016). "On Distances Between KD45n Kripke Models and Their Use for Belief Revision". In: *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*. Ed. by Gal A. Kaminka et al. Vol. 285. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 1053–1061. DOI: 10.3233/978-1-61499-672-9-1053.

Caridroit, Thomas, Jean-Marie Lagniez, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail (2017). "A SAT-Based Approach for Solving the Modal Logic S5-Satisfiability Problem". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*. Ed. by Satinder P. Singh and Shaul Markovitch. AAAI Press, pp. 3864–3870.

van Ditmarsch, Hans, Andreas Herzig, and Tiago de Lima (Nov. 2007a). "Optimal Regression for Reasoning about Knowledge and Actions". In: *Formal Models of Belief Change in Rational Agents*. Ed. by G. Bonanno, J. Delgrande, J. Lang, and H. Rott. Dagstuhl Seminar Proceedings 07351. (Longer version based on contribution 'Optimal Regression for Reasoning about Knowledge and Actions' published in Proceedings of AAAI. 2007). Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

— (2007b). "Optimal Regression for Reasoning about Knowledge and Actions". In: *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI 2007)*. ISBN: 978-1-57735-323-2. AAAI Press, pp. 1070–1075. ISBN: 978-1-57735-323-2.

van Ditmarsch, Hans, Tiago de Lima, and Emiliano Lorini (2011). "Intention change via local assignments". In: *Third international Workshop on Language, Methodologies and Development Tools for Multi-Agent Systems*. Vol. 6822. Lecture Notes in Computer Science. Springer, pp. 135–151.

Herzig, Andreas, Tiago de Lima, and Emiliano Lorini (2009a). "On the Dynamics of Institutional Agreements". In: *Knowledge Representation for Agents and Multi-Agent Systems*. Ed. by J.-J. Meyer and J. Broersen. Vol. 5605. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-05301-6. Springer, pp. 66–80.

Herzig, Andreas, Tiago de Lima, Emiliano Lorini, and Nicolas Troquard (2012). "A Computationally Grounded Dynamic Logic of Agency, with an Application to Legal Actions". In: *Proceedings of the 11th International Conference Deontic Logic in Computer Science (DEON 2012)*. Ed. by Thomas Ågotnes, Jan M. Broersen, and Dag Elgesem. Vol. 7393. Lecture Notes in Computer Science. Springer, pp. 170–183. ISBN: 978-3-642-31569-5.

Lagniez, Jean-Marie, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail (2016). "On Checking Kripke Models for Modal Logic K". In: *Proceedings of the 5th Workshop on Practical Aspects of Automated Reasoning (PAAR@IJCAR 2016)*. Ed. by Pascal Fontaine, Stephan Schulz, and Josef Urban. Vol. 1635. CEUR Workshop Proceedings. CEUR-WS.org, pp. 69–81.

— (2017). "A Recursive Shortcut for CEGAR: Application To The Modal Logic K Satisfiability Problem". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*. Ed. by Carles Sierra. ijcai.org, pp. 674–680. DOI: 10.24963/IJCAI.2017/94.

— (2018b). "An Assumption-Based Approach for Solving the Minimal S5-Satisfiability Problem". In: *Proceedings of the 9th International Joint Conference Automated Reasoning (IJCAR 2018)*. Ed. by Didier Galmiche, Stephan Schulz, and Roberto Sebastiani. Vol. 10900. Lecture Notes in Computer Science. Springer, pp. 1–18. DOI: 10.1007/978-3-319-94205-6_1.

de Lima, Tiago (2011). "Alternating-Time Temporal Announcement Logic". In: *Proceedings of the 12th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA XII)*. Ed. by João Leite, Paolo Torroni, Thomas Ågotnes, Guido Boella, and Leon van der Torre. Vol. 6814. Lecture Notes in Computer Science. Springer, pp. 105–121. ISBN: 978-3-642-22358-7.

de Lima, Tiago, Lambèr Royakkers, and Frank Dignum (2010a). "Modeling the problem of many hands in organisations". In: *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*. Ed. by Helder Coelho, Rudi Studer, and Michael J. Wooldridge. IOS Press, pp. 79–84.

— (July 2008). "Towards a Formalization of Responsibility". In: *Proceedings of the 3rd International Workshop on Normative Multiagent Systems (NorMAS 2008)*. Ed. by G. Boella, M. Singh, G. Pigozzi, and H. Verhagen, pp. 66–79. ISBN: 2-919940-48-1.

Short Papers in International Conference Proceedings

Caridroit, Thomas, Sébastien Konieczny, Tiago de Lima, and Pierre Marquis (2015b). "Private Revision in a Multi-Agent Setting". In: *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*. Ed. by Gerhard Weiss, Pinar Yolum, Rafael H. Bordini, and Edith Elkind. ACM, pp. 1677–1678.

Lagniez, Jean-Marie, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail (2018a). "A SAT-Based Approach For PSPACE Modal Logics". In: *Proceedings of the Sixteenth International Conference Principles of Knowledge Representation and Reasoning (KR 2018)*. Ed. by Michael Thielscher, Francesca Toni, and Frank Wolter. AAAI Press, pp. 651–652.

de Lima, Tiago, Lambèr Royakkers, and Frank Dignum (2009). "Behaving Responsible in Multi-Agent Worlds (Extended Abstract)". In: *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*. Ed. by K. Decker, J. Sichman, C. Sierra, and C. Castelfranchi. IFAAMAS, pp. 1139–1140.

Lorini, Emiliano, Hans van Ditmarsch, and Tiago de Lima (2010). "Logical Model of Intention and Plan Dynamics". In: *Proceedings of the 19th European Conference of Artificial Intelligence (ECAI 2010)*. Ed. by Helder Coelho, Rudi Studer, and Michael Wooldridge. IOS Press, pp. 1075–1076.

Book Chapters

Herzig, Andreas, Tiago de Lima, Emiliano Lorini, and Nicolas Troquard (2015). "Three Traditions in the Logic of Action: Bringing Them Together". In: *Krister Segerberg on Logic of Actions*. Ed. by Robert Trypuz. Vol. 1. Outstanding Contributions to Logic Series. Springer, pp. 61–84.

de Lima, Tiago and Lambèr Royakkers (2015). "A Formalisation of Moral Responsibility and the Problem of Many Hands". In: *Moral Responsibility and the Problem of Many Hands*. Ed. by Ibo van de Poel, Lambèr Royakkers, and Sjoerd D. Swart. Routledge Studies in Ethics and Moral Theory. Taylor & Francis. Chap. 3, pp. 93–130.

References

Abate, Pietro, Rajeev Goré, and Florian Widmann (2007). "Cut-free single-pass tableaux for the logic of common knowledge". In: *Workshop on Agents and Deduction at TABLEAUX 2007*.

Ågotnes, Thomas, Philippe Balbiani, Hans van Ditmarsch, and Pablo Seban (2010). "Group announcement logic". In: *Journal of Applied Logic* 8.1, pp. 62–81.

Ågotnes, Thomas and Hans van Ditmarsch (2008). "Coalitions and Announcements". In: *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*. Ed. by L. Padgham, D. C. Parkes, J. P. Müeller, and S. Persons. Vol. 2. IFAAMAS, pp. 673–680.

Alchourrón, Carlos E., Peter Gärdenfors, and David Makinson (1985). "On the logic of theory change: Partial meet contraction and revision functions". In: *Journal of Symbolic Logic* 50, pp. 510–530.

Alechina, Natasha, Joseph Y. Halpern, and Logan Brian (2017). "Causality, responsibility and blame in team plans". In: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*. Ed. by Kate Larson, Michael Winikoff, Sanmay Das, and Edmund H. Durfee. IFAAMAS, pp. 1091–1099.

d'Altan, P., John-Jules Ch. Meyer, and Roel J. Wieringa (1996). "An Integrated Framework for Ought-to-Be and Ought-to-Do Constraints". In: *Artificial Intelligence and Law* 4.2, pp. 77–111.

Alur, Rajeev, Thomas A. Henzinger, and Orna Kupferman (2002). "Alternating-time temporal logic". In: *Journal of the ACM* 5.49, pp. 672–713.

Aucher, Guillaume (2008). "Perspectives on belief and change". PhD thesis. Université Paul Sabatier; University of Otago.

— (2010). "Generalizing AGM to a multi-agent setting". In: *Logic Journal of the IGPL* 18.4, pp. 530–558.

— (2012). "Private announcement and belief expansion: an internal perspective". In: *Journal of Logic and Computation* 22.3, pp. 451–479.

Audemard, Gilles, Jean-Marie Lagniez, and Laurent Simon (2013). "Improving Glucose for Incremental SAT Solving with Assumptions: Application to MUS Extraction". In: *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing (SAT 2013)*. Vol. 7962. Lecture Notes in Computer Science. Springer, pp. 309–317.

Audemard, Gilles and Laurent Simon (2009). "Predicting Learnt Clauses Quality in Modern SAT Solvers". In: *Proceedings of the 21th International Joint Conference on Artificial Intelligence (IJCAI 2009)*. IJCAI Organization, pp. 399–404.

Baaz, Matthias, Uwe Egly, and Alexander Leitsch (2001). "Normal Form Transformations". In: *Handbook of Automated Reasoning*. Ed. by John Alan Robinson and Andrei Vronkov. Vol. 1. MIT Press. Chap. 5, pp. 273–333.

Balbiani, Philippe, Alexandru Baltag, Hans van Ditmarsch, Andreas Herzig, Tomohiro Hoshi, and Tiago de Lima (Oct. 2008). "'Knowable' as 'known after an announcement'". In: *The Review of Symbolic Logic* 1.3, pp. 305–334. DOI: 10.1017/S1755020308080210.

Balbiani, Philippe, Hans van Ditmarsch, Andreas Herzig, and Tiago de Lima (2012). "Some truths are best left unsaid". In: *Advances in Modal Logic Volume 9*. Ed. by Thomas Bolander, Torben Braüner, Silvio Ghilardi, and Lawrence S. Moss. College Publications, pp. 36–54.

Balsiger, Peter, Alain Heuerding, and Stefan Schwendimann (2000). "A Benchmark Method for the Propositional Modal Logics K, KT, S4". In: *Journal of Automated Reasoning* 24.3, pp. 297–317.

Baltag, Alexandru. and Sonja Smets (2006). "Dynamic belief revision over multi-agent plausibility models". In: *Proceedings of the 7th Conference on Logic and the Foundations of Game and Decision Theory (LOFT 2006)*, pp. 11–24.

Baltag, Alexandru and Lawrence S. Moss (2004). "Logics for Epistemic Programs". In: *Synthese* 139.2. Knowledge, Rationality & Action 1–60, 2004, pp. 165–224.

Baltag, Alexandru, Lawrence S. Moss, and Slawomir Solecki (1998). "The logic of common knowledge, public announcements and private suspicions". In: *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 1998)*. Ed. by Itzhak Gilboa. Morgan Kaufmann, pp. 43–56.

Bayardo Jr, Roberto J. and Robert Schrang (1997). "Using CSP look-back techniques to solve real world SAT instances". In: *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI 1997)*. The AAAI Press, pp. 203–208.

Belnap, Nuel, Michael Perloff, and Ming Xu (2001). *Facing the Future: Agents and Choices in Our Indeterminist World*. Oxford University Press.

Benferhat, Salem, Zied Bouraoui, Odile Papini, and Eric Würbel (2014). "Assertional-based Prioritized Removed Sets Revision of DL-LiteR Knowledge Bases". In: *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*. Ed. by Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan. Vol. 263. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 967–968. DOI: 10.3233/978-1-61499-419-0-967.

— (2017). "Prioritized assertional-based removed sets revision of DL-Lite belief bases". In: *Annals of Mathematics and Artificial Intelligence* 79.1-3, pp. 45–75. DOI: 10.1007/S10472-015-9494-2.

van Benthem, Johan (1984). "Correspondence Theory". In: *Handbook of Philosophical Logic: Extensions of Classical Logic*. Ed. by Dov M. Gabbay and Franz Guenther. Vol. 2. D. Reidel Publishing Company. Chap. 4, pp. 167–247.

— (2007). "Dynamic logic for belief revision". In: *Journal of Applied Non-Classical Logics* 17.2, pp. 129–155.

Biere, Armin, Marijn Heule, Hans van Maaren, and Toby Walsh, eds. (2009). *Handbook of Satisfiability*. Vol. 185. Frontiers in Artificial Intelligence and Applications. IOS Press. ISBN: 978-1-58603-929-5.

Blackburn, Patrick, Johan van Benthem, and Frank Wolter, eds. (2007). *Handbook of Modal Logic*. Vol. 3. Studies in Logic and Practical Reasoning. New York, NY, USA: Elsevier Science Inc. ISBN: 0444516905.

Blackburn, Patrick, Maarten de Rijke, and Yde Venema (2001). *Modal Logic*. Cambridge University Press.

Board, Oliver (2004). "Dynamic interactive epistemology". In: *Games and Economic Behavior* 49.1, pp. 49–80.

Borgo, Stefano (2007). "Coalitions in Action Logic". In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*. Ed. by Manuela M. Veloso, pp. 1822–1827.

Bovens, Mark (1998). *The quest for responsibility. Accountability and citizenship in complex organisations*. Cambridge University Press.

Broersen, Jan M. (2011). "Deontic epistemic stit logic distinguishing modes of mens rea". In: *Journal of Applied Logic* 9, pp. 137–152.

Broersen, Jan M., Andreas Herzig, and Nicolas Troquard (2007). "A Normal Simulation of Coalition Logic and an Epistemic Extension". In: *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK XI)*. Ed. by Dov Samet, pp. 92–101.

Brummayer, Robert and Armin Biere (2009). "Effective Bit-Width and Under-Approximation". In: *Proceedings of 12th International Conference on Computer Aided Systems Theory (EUROCAST 2009)*. Ed. by Roberto Moreno-Díaz, Franz Pichler, Quesada-Arencibia, and Alexis. Vol. 5717. Lecture Notes in Computer Science. Springer, pp. 304–311.

Bryant, Randal E. (1986). "Graph-Based Algorithms for Boolean Function Manipulation". In: *IEEE Transactions on Computers* 35.8, pp. 677–691.

Caridroit, Thomas, Sébastien Konieczny, Tiago de Lima, and Pierre Marquis (2015a). "Private Expansion and Revision in Multi-Agent Settings". In: *Proceedings of the 13th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2015)*. Ed. by Sébastien Destercke and Thierry Denoeux. Vol. 9161. Lecture Notes in Computer Science. Springer, pp. 175–185.

— (2015b). "Private Revision in a Multi-Agent Setting". In: *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*. Ed. by Gerhard Weiss, Pinar Yolum, Rafael H. Bordini, and Edith Elkind. ACM, pp. 1677–1678.

— (2016). "On Distances Between KD45n Kripke Models and Their Use for Belief Revision". In: *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*. Ed. by Gal A. Kaminka et al. Vol. 285. Frontiers in Artificial Intel-

ligence and Applications. IOS Press, pp. 1053–1061. DOI: 10.3233/978-1-61499-672-9-1053.

Caridroit, Thomas, Jean-Marie Lagniez, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail (2017). "A SAT-Based Approach for Solving the Modal Logic S5-Satisfiability Problem". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*. Ed. by Satinder P. Singh and Shaul Markovitch. AAAI Press, pp. 3864–3870.

Chellas, Brian F. (1980). *Modal Logic: an introduction*. Cambridge University Press.

Chockler, Hana and Joseph Y. Halpern (2004). "Responsibility and Blame: A Structural-Model Approach". In: *Journal of Artificial Intelligence Research* 22, pp. 93–115.

Clarke, Edmund M., Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veit (2003). "Counterexample-guided abstraction refinement for symbolic model checking". In: *Journal of the ACM* 50.5, pp. 752–794.

Cook, Stephen A. (1971). "The complexity of theorem-proving procedures". In: *Proceedings of the 3rd annual ACM symposium on Theory of computing (STOC 1971)*. ACM, pp. 151–158.

Creignou, Nadia, Raïda Ktari, and Odile Papini (2016). "Belief Contraction Within Fragments of Propositional Logic". In: *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*. Vol. 285. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 390–398. DOI: 10.3233/978-1-61499-672-9-390.

Creignou, Nadia, Odile Papini, Reinhard Pichler, and Stefan Woltran (2014). "Belief revision within fragments of propositional logic". In: *Journal of Computer and System Science* 80.2, pp. 427–449. DOI: 10.1016/J.JCSS.2013.08.002.

D'Agostino, Marcello, Dov M. Gabbay, Reiner Hähnle, and Joachim Possega, eds. (1999). *Handbook of Tableau Methods*. Kluwer Academic Publishers.

Davis, Martin, George Logemann, and Donald Loveland (1962). "A Machine Program for Theorem Proving". In: *Communications of the ACM* 5.7, pp. 394–397.

Davis, Martin and Hilary Putnam (1960). "A Computing Procedure for Quantification Theory". In: *Journal of the ACM* 7.3, pp. 201–215.

De Bona, Glauber, John Grant, Anthony Hunter, and Sébastien Konieczny (2018). "Towards a Unified Framework for Syntactic Inconsistency Measures". In: *Proceedins of the 22nd AAAI Conference on Artificial Intelligence (AAAI 2018)*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, pp. 1803–1810.

Delgrande, James P. and Pavlos Peppas (2011). "Revising Horn Theories". In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*. Ed. by Toby Walsh. IJCAI/AAAI, pp. 839–844. DOI: 10.5591/978-1-57735-516-8/IJCAI11-146.

Demolombe, Robert, Andreas Herzig, and Ivan J. Varzinczak (2003). "Regression in Modal Logic". In: *Journal of Applied Non-Classical Logics* 13.2, pp. 165–185.

Dignum, Virginia and Frank Dignum (2007). "A Logic for Agent Organizations". In: *Proceedings of the 3rd Workshop on Formal Approaches to Multi-Agent Systems (FAMAS 2007)*.

van Ditmarsch, Hans (2005). "Prolegomena to Dynamic Logic for Belief Revision". In: *Synthese* 147.2, pp. 229–275.

van Ditmarsch, Hans, Joseph Y. Halpern, Wiebe van der Hoek, and Barteld Kooi, eds. (2015). *Handbook of Epistemic Logic*. College Publications.

van Ditmarsch, Hans, Andreas Herzig, and Tiago de Lima (2007). "Optimal Regression for Reasoning about Knowledge and Actions". In: *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI 2007)*. ISBN: 978-1-57735-323-2. AAAI Press, pp. 1070–1075. ISBN: 978-1-57735-323-2.

van Ditmarsch, Hans, Wiebe van der Hoek, and Barteld Kooi (2005). "Dynamic epistemic logic with assignment". In: *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*. Ed. by Frank Dignum, Virginia Dignum, Koenig S., S. Kraus, M. P. Sing, and Michael Wooldridge. ACM, pp. 141–148.

— (2007). *Dynamic Epistemic Logic*. Springer.

Dyckhoff, Roy, ed. (2000). Vol. 1847. Lecture Notes in Computer Science. Springer.

Eén, Niklas and Niklas Sörensson (2003). "An Extensible SAT-solver". In: *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing (SAT 2003)*. Vol. 2929. Lecture Notes in Computer Science. Springer, pp. 502–518.

Fagin, Ronald, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi (1995). *Reasoning about Knowledge*. The MIT Press.

Fitting, Melvin (Aug. 2010). "Notes on Classical Propositional Logic". Notes for 'Non-Classical Logic' course, Phil 76500 Spring 2018, Lehman College, CUNY, USA.

van Fraassen, Bas C. (1973). "Values and the heart's command". In: *The Journal of Philosophy* 70.1, pp. 5–19.

French, Tim and Hans van Ditmarsch (Sept. 2008). "Undecidability for arbitrary public announcement logic". In: *Advances in Modal Logic 7, papers from the seventh conference on "Advances in Modal Logic"*. Ed. by Carlos Areces and Robert Goldblatt. Nancy, France: College Publications, pp. 23–42.

Gabbay, Dov M. and Franz Guenther, eds. (1984). *Handbook of Philosophical Logic: Extensions of Classical Logic*. Vol. 2. D. Reidel Publishing Company.

Gärdenfors, Peter (2008). *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. Ed. by Dov M. Gabbay, J. Siekmann, Johan van Benthem, and J. Woods. Vol. 13. Logic and Cognitive Systems. College Publications.

Gerbrandy, Jelle (May 2006). "Logics of propositional control". In: *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*. Ed. by Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone. Hakodate, Japan: ACM, pp. 193–200.

Giunchiglia, Enrico and Armando Tacchella (2000a). "A subset-matching size-bounded cache for satisfiability in modal logics". In: *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2000)*. Ed. by Roy Dyckhoff. Vol. 1847. Lecture Notes in Computer Science. Springer, pp. 237–251.

— (2000b). "System description: *SAT: A platform for the development of modal decision procedures". In: *Proceedings of 17th International Conference on Automated Deduction (CADE-17)*. Ed. by David McAllester. Vol. 1831. Lecture Notes in Computer Science. Springer, pp. 291–296.

Giunchiglia, Enrico, Armando Tacchella, and Fausto Giunchiglia (2002). "SAT-Based Decision Procedures for Classical Modal Logics". In: *Journal of Automated Reasoning* 28.2, pp. 143–171.

Goldman, Alvin and Thomas Blanchard (2018). "Social Epistemology". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2018. Metaphysics Research Lab, Stanford University.

Goodin, Robert E. (1995). *Utilitarianism as a Public Philosophy*. Cambridge University Press.

Goranko, Valentin and Govert van Drimmelen (Mar. 2006). "Complete axiomatization and decidability of Alternating-time temporal logic". In: *Theoretical Computer Science* 353.1–3, pp. 93–117. DOI: 10.1016/j.tcs.2005.07.043.

Goré, Rajeev, Kerry Olesen, and Jimmy Thomson (2014). "Implementing Tableau Calculi Using BDDs: BDDTab System Description". In: *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2014)*. Ed. by Stéphane Demri, Deepak Kapur, and Christoph Weidenbach. Vol. 8562. Lecture Notes in Computer Science. Springer, pp. 337–343.

Goré, Rjeev (1999). "Tableau Methods for Modal and Temporal Logics". In: *Handbook of Tableau Methods*. Ed. by Marcello D'Agostino, Dov M. Gabbay, Reiner Hähnle, and Joachim Possega. Kluwer Academic Publishers, pp. 297–396.

Götzmann, Daniel, Mark Kaminski, and Gert Smolka (2010). "Spartacus: A Tableau Prover for Hybrid Logic". In: *Electronic Notes in Theorethical Computer Science* 262, pp. 127–139.

Grossi, Davide, Lambèr Royakkers, and Frank Dignum (2007). "Organizational structure and responsibility: An analysis in a dynamic logic of organized collective agency". In: *Artificial Intelligence and Law* 15, pp. 223–249.

Grove, Adam (1988). "Two Modellings of Theory Change". In: *Journal of Philosophical Logic* 17.2, pp. 157–170.

Halpern, Joseph Y. (2015). "A modification of the Halpern-Pearl definition of causality." In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pp. 3022–3033.

Halpern, Joseph Y. and Yoram Moses (Apr. 1992). "A guide to completeness and complexity for modal logics of knowledge and belief". In: *Artificial Intelligence* 54.3, pp. 319–379.

Halpern, Joseph Y. and Leandro Chaves Rêgo (Jan. 2007a). "Characterizing the NP-PSPACE Gap in the Satisfiability Problem for Modal Logic". In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*. Ed. by Manuela M. Veloso, pp. 2306–2311.

— (Aug. 2007b). "Characterizing the NP-PSPACE Gap in the Satisfiability Problem for Modal Logic". In: *Journal of Logic and Computation* 17.4, pp. 795–806.

Harel, David (1984). "Dynamic Logic". In: *Handbook of Philosophical Logic: Extensions of Classical Logic*. Ed. by Dov M. Gabbay and Franz Guenther. Vol. 2. D. Reidel Publishing Company. Chap. 10, pp. 497–604.

Harel, David, Dexter Kozen, and Jerzy Tiuryn (2000). *Dynamic Logic*. MIT Press.

Harper, William L. (1976). "Rational Conceptual Change". In: *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association 1976*. Vol. 2, pp. 462–494.

Herzig, Andreas, Jérôme Lang, Dominique Longin, and Thomas Polacsek (2000). "A logic for planning under partial observability". In: *Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on on Innovative Applications of Artificial Intelligence (AAAI 2000)*. Ed. by Henry A. Kautz and Bruce W. Porter. AAAI Press / The MIT Press, pp. 768–773.

Herzig, Andreas, Jérôme Lang, and Pierre Marquis (2005). "Action progression and revision in multiagent belief structures". In: *Proceedings of the 6th Workshop on Nonmonotonic Reasoning, Action, and Change (NRAC 2005)*.

Herzig, Andreas, Tiago de Lima, and Emiliano Lorini (2009). "On the dynamics of institutional agreements". In: *Synthese* 171, pp. 321–355. DOI: 10.1007/s11229-009-9645-2.

Herzig, Andreas and Emiliano Lorini (Oct. 2010a). "A Dynamic Logic of Agency I: STIT, Capabilities and Powers". In: *Journal of Logic, Language and Information* 19.1, pp. 89–121. DOI: 10.1007/s10849-009-9105-x.

— (July 2010b). "A Dynamic Logic of Agency II: Deterministic , Coalition Logic, and Game Theory". In: *Journal of Logic, Language and Information* 19.3, pp. 327–351. DOI: 10.1007/s10849-009-9104-y.

Hintikka, Jaakko (1962). *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press.

van der Hoek, Wiebe and Michael Wooldridge (2003). "Cooperation, Knowledge, and Time: Alternating-Time Temporal Epistemic Logic and its Applications". In: *Studia Logica* 75, pp. 125–157.

Horrocks, Ian, Ullrich Hustadt, Ulrike Sattler, and Renate A. Schmidt (2007). "4 Computational modal logic". In: *Studies in Logic and Practical Reasoning* 3, pp. 181–245.

Hugues, G.E. and M.J. Cresswell (1996). *A New Introduction to Modal Logic*. Routlege. DOI: 10.4324/9780203028100.

Hunter, Anthony and Sébastien Konieczny (2010). "On the measure of conflicts: Shapley Inconsistency Values". In: *Artificial Intelligence* 174.14, pp. 1007–1026. DOI: 10.1016/J.ARTINT.2010.06.001.

Jamroga, Wojtec and Thomas Ågotnes (2007). "Constructive knowledge: What agents can achieve under incomplete information". In: *Journal of Applied Non-Classical Logics* 4.1, pp. 423–475.

Jamroga, Wojtec and Wiebe van der Hoek (2004). "Agents that know how to play". In: *Fundamenta Informaticae*.

Janota, Mikolás, William Klieber, João Marques-Silva, and Edmund M. Clarke (2016). "Solving QBF with counterexample guided refinement". In: *Artificial Intelligence* 234, pp. 1–25.

Kaminski, Mark and Tobias Tebbi (2013). "InKreSAT: Modal Reasoning via Incremental Reduction to SAT". In: *Proceedings of 24th International Conference on Automated Deduction (CADE-24)*. Ed. by Maria Paola Bonacina. Vol. 7898. Lecture Notes in Computer Science. Springer, pp. 436–442.

Katsuno, Hirofumi and Alberto O. Mendelzon (1991a). "On the difference between updating a knowledge base and revising it". In: *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR 1991)*.

Ed. by James F. Allen, Richard Fikes, and Erik Sandewall. Morgan Kaufmann, pp. 387–394.

Katsuno, Hirofumi and Alberto O. Mendelzon (Dec. 1991b). "Propositional knowledge base revision and minimal change". In: *Artificial Intelligence* 52.3, pp. 263–294. DOI: 10.1016/0004-3702(91)90069-V.

Khasidashvili, Zurab, Konstantin Korovin, and Dmitry Tsarkov (2015). "EPR-based k-induction with Counterexample Guided Abstraction Refinement". In: *Global Conference on Artificial Intelligence (GCAI 2015)*. Ed. by Georg Gottlob, Geoff Sutcliffe, and Andrei Voronkov. Vol. 36. EPiC Series in Computing. EasyChair.

Kooi, Barteld (2007). "Expressivity and completeness for public update logics via reduction axioms". In: *Journal of Applied Non-Classical Logics* 17.2, pp. 231–253.

Kornhauser, Lewis A. and Lawrence G. Sager (1986). "Unpacking the Court". In: *Yale Law Journal* 96, pp. 82–117.

Kovács, Laura and Andrei Voronkov (2013). "First-Order Theorem Proving and Vampire". In: *Proceedings of 25th International Conference on Computer Aided Verification (CAV 2013)*. Ed. by Natasha Sharygina and Helmut Veith. Vol. 8044. Lecture Notes in Computer Science. Springer, pp. 1–35.

Kripke, Saul (1959). "A Completeness Theorem in Modal Logic". In: *Journal of Symbolic Logic* 24.1, pp. 1–14. DOI: 10.2307/2964568.

Kuhn, Steven (2019). "Prisoner's Dilemma". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2019. Metaphysics Research Lab, Stanford University.

Ladner, Richard E. (1977). "The Computational Complexity of Provability in Systems of Modal Propositional Logic". In: *SIAM Journal on Computing* 6.3, pp. 467–480.

Lagniez, Jean-Marie, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail (2017). "A Recursive Shortcut for CEGAR: Application To The Modal Logic K Satisfiability Problem". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*. Ed. by Carles Sierra. ijcai.org, pp. 674–680. DOI: 10.24963/IJCAI.2017/94.

— (2018). "A SAT-Based Approach For PSPACE Modal Logics". In: *Proceedings of the Sixteenth International Conference Principles of Knowledge Representation and Reasoning (KR 2018)*. Ed. by Michael Thielscher, Francesca Toni, and Frank Wolter. AAAI Press, pp. 651–652.

Lakemeyer, Gerhard and Hector J. Levesque (2005). "Semantics for a useful fragment of the situation calculus". In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*. Ed. by L. P. Kaelbling and A. Saffiotti. Professional Book Center, pp. 490–496.

Lemmon, Edward J. and Dana S. Scott (1977). *The Lemmon Notes: An Introduction to Modal Logic*. Ed. by Krister Segerberg. Vol. 11. American Philosophical Quarterly Monograph Series. Oxford: Basil Blackwell.

Levi, Isaac (1977). "Subjunctives, Dispositions and Chances". In: *Synthese* 34, pp. 423–455.

de Lima, Tiago (2011). "Alternating-Time Temporal Announcement Logic". In: *Proceedings of the 12th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA XII)*. Ed. by João Leite, Paolo Torroni, Thomas Ågotnes, Guido

Boella, and Leon van der Torre. Vol. 6814. Lecture Notes in Computer Science. Springer, pp. 105–121. ISBN: 978-3-642-22358-7.

— (2014). "Alternating-time temporal dynamic epistemic logic". In: *Journal of Logic and Computation* 24.6, pp. 1145–1178. DOI: 10.1093/logcom/exs061.

de Lima, Tiago, Lambèr Royakkers, and Frank Dignum (2010a). "Modeling the problem of many hands in organisations". In: *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*. Ed. by Helder Coelho, Rudi Studer, and Michael J. Wooldridge. IOS Press, pp. 79–84.

de Lima, Tiago and Lambèr Royakkers (2015). "A Formalisation of Moral Responsibility and the Problem of Many Hands". In: *Moral Responsibility and the Problem of Many Hands*. Ed. by Ibo van de Poel, Lambèr Royakkers, and Sjoerd D. Swart. Routledge Studies in Ethics and Moral Theory. Taylor & Francis. Chap. 3, pp. 93–130.

de Lima, Tiago, Lambèr Royakkers, and Frank Dignum (2010b). "A Logic for Reasoning about Responsibility". In: *Logic Journal of the IGPL* 18.1, pp. 99–117. DOI: 10.1093/jigpal/jzp073.

Magnier, Sébastien and Tiago de Lima (June 2015). "A soundness & completeness proof on dialogs and dynamic epistemic logic". In: *Logique & Analyse* 230, pp. 219–250. DOI: 10.2143/LEA.230.0.3141809.

Marques-Silva, João, Ines Lynce, and Sharad Malik (2009). "Conflict-Driven Clause Learning SAT Solvers". In: *Handbook of Satisfiability*. Ed. by Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh. IOS Press. Chap. 4, pp. 131–153.

Marques-Silva, João and Karem A. Sakallah (1996). "GRASP–A New Search Algorithm for Satisfiability". In: *Digest of IEEE International Conference on Computer-Aided Design (ICCAD 1996)*, pp. 220–227.

— (1999). "GRASP: A Search Algorithm for Propositional Satisfiability". In: *IEEE Transactions on Computers* 48.5, pp. 506–521.

Massacci, Fabio (1999). "Design and Results of the Tableaux-99 Non-classical (Modal) Systems Comparison". In: *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 1999)*. Ed. by Neil V. Murray. Vol. 1617. Lecture Notes in Computer Science. Springer, pp. 14–18.

Massacci, Fabio and Francesco M. Donini (2000). "Design and Results of TANCS-2000 Non-classical (Modal) Systems Comparison". In: *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2000)*. Ed. by Roy Dyckhoff. Vol. 1847. Lecture Notes in Computer Science. Springer, pp. 52–56. DOI: 10.1007/10722086_4.

Mendelson, Elliot (2015). *An Introduction to Mathematical Logic*. Taylor & Francis.

Meyer, John-Jules Ch. (1988). "A Different Approach to Deontic Logic: Deontic Logic Viewed as a Variant of Dynamic Logic". In: *Notre Dame Journal of Formal Logic* 29.1, pp. 109–136.

Meyer, John-Jules Ch. and Wiebe van der Hoek (1995). *Epistemic Logic for AI and Computer Science*. Cambridge University Press.

Meyer, John-Jules Ch. and Roel J. Wieringa, eds. (1993). *Deontic Logic in Computer Science*. Chichester: John Wiley and Sons.

Montmirail, Valentin (Sept. 2018). "Practical resolution of satisfiabilitytesting for modal logics". PhD thesis. Lens, France: Artois University.

Mu, Kedian, Weiru Liu, and Zhi Jin (2011). "A general framework for measuring inconsistency through minimal inconsistent sets". In: *Knowledge and Information Systems* 27.1, pp. 85–114. DOI: 10.1007/S10115-010-0295-Y.

Nalon, Cláudia, Ullrich Hustadt, and Clare Dixon (2016). "K_SP : A Resolution-Based Prover for Multimodal K". In: *Proceedings International Joint Conference on Automated Reasoning (IJCAR 2016)*. Ed. by Nicola Olivetti and Ashish Tiwari. Vol. 9706. Lecture Notes in Computer Science. Springer, pp. 406–415.

Osborne, Martin J. and Ariel Rubinstein (1994). *A Course in Game Theory*. The MIT Press.

Patel-Schneider, Peter F. and Roberto Sebastiani (Apr. 2003). "A New General Method to Generate Random Modal Formulae for Testing Decision Procedures". In: *Journal of Artificial Intelligence Research* 18, pp. 351–389. DOI: 10.1613/jair.1166.

Pauly, Marc (2001). "Logic for Social Software". PhD thesis. ILLC, University of Amsterdam.

— (2002). "A Modal Logic for Coalitional Power in Games". In: *Journal of Logic and Computation* 12.1, pp. 149–166.

Petit, Philip (2001). "Deliberative Democracy, the Discursive Dilemma". In: *Philosophical Issues* 11, pp. 268–299.

Plaza, Jan A. (1989). "Logics of public communications". In: *Proceedings of the fourth international symposium on methodologies for intelligent systems: Poster session program (ISMIS 1989)*. Ed. by M. L. Emrich, M. S. Pfeifer, M. Hadzikadic, and Z. W. Ras. Oak Ridge National Laboratory, pp. 201–216.

van de Poel, Ibo, Lambèr Royakkers, and Sjoerd D. Swart, eds. (2015). *Moral Responsibility and the Problem of Many Hands*. Routledge Studies in Ethics and Moral Theory. Taylor & Francis.

Reiter, Raymond (1991). "The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression". In: *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*. Ed. by Vladimir Lifschitz. New York: Academic Press, pp. 359–380.

Robinson, John Alan (1965). "A Machine-Oriented Logic Based on the Resolution Principle". In: *Journal of the ACM* 12.1, pp. 23–41.

Royakkers, Lambèr (1998). *Extending Deontic Logics for the Formalisation of Legal Rules*. Kluwer Academic Publishers.

Sahlqvist, Henrik (1975). "Completeness and Correspondence in the First and Second Order Semantics for Modal Logic". In: *Proceedings of the Third Scandinavian Logic Symposium. Uppsala 1973*. Ed. by Stig Kanger. North-Holland Publishing Company, pp. 110–143.

Santos, Felipe and José Carmo (1996). "Indirect Action, Influence and Responsibility". In: *Deontic Logic, Agency and Normative Systems*. Ed. by M. A. Brown and J. Carmo. C. J. van Rijsbergen, editor, Workshops in Computing series. Springer.

Sebastiani, Roberto and Armando Tacchella (2009). "SAT Techniques for Modal and Description Logics". In: *Handbook of Satisfiability*. Ed. by Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh. Vol. 185. Frontiers in Artificial Intelligence and Applications. IOS Press. Chap. 25, pp. 781–824. DOI: 10.3233/978-1-58603-929-5-781.

Sebastiani, Roberto and Michele Vescovi (2009). "Automated Reasoning in Modal and Description Logics via SAT Encoding: the Case Study of K(m)/ALC-Satisfiability". In: *Journal of Artificial Intelligence Research* 35.1, pp. 343–389.

Seipp, Jendrik and Malte Helmert (2013). "Counterexample-Guided Cartesian Abstraction Refinement". In: *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS 2013)*. Ed. by Daniel Borrajo, Subbarao Kambhampati, Angelo Oddi, and Simone Fratini. AAAI Press.

Tallon, Jean-Marc, Jean-Christophe Vergnaud, and Shmuel Zamir (2004). "Communication among Agents: A Way to Revise Beliefs in KD45 Kripke Structures." In: *Journal of Applied Non-Classical Logics* 14.4, pp. 477–500.

Thompson, Dennis F. (1980). "Moral responsibility and public officials: The problem of many hands". In: *American Political Science Review* 74.4, pp. 905–916.

Tsarkov, Dmitry and Ian Horrocks (2006). "FaCT++ Description Logic Reasoner: System Description". In: *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2006)*. Ed. by Ulrich Furbach and Natarajan Shankar. Vol. 4130. Lecture Notes in Computer Science. Springer, pp. 292–297.

Tseitin, G. S (1983). "On the Complexity of Derivation in Propositional Calculus". In: *Automation of Reasoning 2: Classical Papers on Computational Logic 1967–1970*. Ed. by Jörg H. Siekmann and Graham Wrightson. Springer, pp. 466–483.

Walther, Dirk, Wiebe van der Hoek, and Michael Wooldridge (2007). "Alternating-time Temporal Logic with Explicit Strategies". In: *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK XI)*. Ed. by D. Samet. Presses Universitaires de Louvain, pp. 269–278.

Wang, Chao, Aarti Gupta, and Franjo Ivancic (2007). "Induction in CEGAR for Detecting Counterexamples". In: *Proceedings of the 7th International Conference on Formal Methods in Computer-Aided Design (FMCAD 2007)*. IEEE Computer Society, pp. 77–84.

Weidenbach, Christoph, Dilyana Dimova, Arnaud Fietzke, Rohit Kumar, Martin Suda, and Patrick Wischnewski (2009). "SPASS Version 3.5". In: *Proceedings of 22nd International Conference on Automated Deduction (CADE-22)*. Ed. by Renate A. Schmidt. Vol. 5663. Lecture Notes in Computer Science. Springer, pp. 140–145.

Wieringa, Roel J. and John-Jules Ch. Meyer (1993). "Actors, actions, and initiative in normative system specification". In: *Annals of Mathematics and Artificial Intelligence* 7, pp. 289–346.

Alphabetical Index

Résumé

Ce travail présente quatre des mes résultats de recherche les plus importants obtenus après ma thèse de doctorat, c'est-à-dire, de 2008 à 2019. Après un bref état de l'art, un formalisme visant à modéliser la responsabilité dans des environnements multi-agents est mis en avant. Ce formalisme, appelé CEDL, est une logique modale qui contient des opérateurs épistémiques et dynamiques ainsi que d'autres opérateurs définis comme des abréviations, telles que l'obligation, la capacité des agents et la capacité de savoir des agents. Le deuxième résultat est une évolution de CEDL, appelée ATDEL. Cette logique vise à modéliser les capacités et les connaissances des agents dans un cadre temporel. Elle permet des spécifications des systèmes beaucoup plus courtes que celles de CEDL. De plus, outre une axiomatisation, des algorithmes de vérification de modèles et de vérification de la satisfiabilité de formules sont proposés. Le troisième résultat est une adaptation de la théorie de révision de croyances AGM à des scénarios multi-agents. Les postulats d'expansion et de révision AGM sont généralisés à de tels scénarios et des opérateurs concrets pour l'expansion et la révision des croyances multi-agents sont proposés. Le dernier résultat présenté porte sur le raisonnement automatique pour les logiques modales. Des méthodes de vérification de la satisfiabilité de formules dans les logiques modales K et KT5 sont proposés. Des expériences pratiques montrent que notre techniques surpasse toutes les méthodes alternatives.

Mots-clés: intelligence artificielle · représentation de connaissance et raisonnement · systèmes multi-agents · logique modale · raisonnement sur les actions · raisonnement sur la connaissance et la croyance · responsabilité · capacité des agents · révision de croyances · raisonnement automatique · satisfiabilité

Abstract

This work presents four of the most important research results obtained after my Ph.D., i.e., from 2008 until 2019. After a brief state of the art, a formalism aiming at modelling responsibility in multi-agent environments is put forward. This formalism, called CEDL, is a modal logic that contains epistemic and dynamic operators as well as some other operators defined as abbreviations, such as obligation, agents abilities and agents knowing how abilities. The second result is an evolution of CEDL, called ATDEL. This logic aims at modelling agents abilities and knowledge trough time. It permits system specifications that are much smaller than CEDL. In addition, apart from an axiomatisation, algorithms for model checking and satisfiability checking of formulas are also provided. The third result is an adaptation of AGM belief revision theory to multi-agent scenarios. Multi-agent versions of expansion and revision postulates are proposed. The last result presented in this work is about modal logic automated reasoning. Methods for satisfiability checking of formulas in K and KT5 are proposed. Practical experiments show that our technique outperforms alternative approaches.

Keywords: artificial intelligence · knowledge representation and reasoning · multi-agent systems · modal logic · reasoning about actions · reasoning about knowledge and belief · responsibility · agent abilities · belief revision · automated reasoning · satisfiability