A Logic of Finite Syntactic Epistemic States

Thomas Ågotnes

Doctor Scientiarum Dissertation

April 2004

Department of Informatics
University of Bergen
Norway

Abstract

The thesis presents a logic of the explicit knowledge of deliberative agents who represent their knowledge symbolically as sets of formulae – agents with finite syntactic epistemic states. It is well known that modal epistemic logic either describes implicit knowledge, including all logical consequences of the explicit knowledge, or describes the explicit knowledge of unrealistically intelligent and powerful agents, but does not describe the explicit knowledge of real agents. A source of this problem is the failure to separate the concepts of explicit knowledge and reasoning. The logic in this thesis consists of two parts. The first is a logic of explicit syntactic knowledge, with no closure conditions, which can be viewed as knowledge at a given point in time. Although this is a simple concept, the meta-language is expressive enough to allow the development of an interesting theory of static finite syntactic epistemic states. The second part is a logic of the evolving explicit syntactic knowledge of agents who have reasoning and communication mechanisms in addition to syntactical storage, i.e. about how the epistemic states can change over time as a result of reasoning and/or communication. The language introduces expressions for knowing a rule, analogue to knowing a formula. Instead of the usual closure conditions on knowledge it is possible to express the fact that if an agent knows some formulae and he knows a certain rule, then he may get to know a conclusion if he chooses to. This model is based on Alternating-time Temporal Logic (ATL). It differs from related epistemic logics based on ATL in that it models explicit instead of implicit knowledge and in the expressiveness of rule operators. It differs from related models of evolving explicit knowledge in that it allows reasoning in terms of possible futures, cooperation and strategies.

Acknowledgments

Michał Walicki has been my supervisor during the work with this dissertation. I am grateful for the great interest he has shown my work, and for the influence he has had on my understanding of logic and computer science.

Hans K. Hvide provided help and motivation in the beginning of the project. Arild Waaler showed an interest in the work at an important stage, which was a great encouragement. I have also had valuable discussions directly related to aspects of the dissertation with Marc Bezem and Uwe Wolter.

Thank you.

CONTENTS

Part I

Preliminaries

Chapter 1

Introduction

1.1 Reasoning About Knowledge

Formal methods for reasoning about knowledge are used to design and analyze artificial intelligent agents, to design and analyze distributed computer systems, to predict the outcome of games with rational players, to provide formal semantics for natural languages, and by philosophers to gain a better understanding of the concept of knowledge.

In artificial intelligence and computer science, the word "knowledge" is most often used as a metaphor. For example, we may say that a car wash machine *knows* the position of a car by the use of optical sensors, or that a web browser *knows* the previous page you visited. For a slightly more advanced example, if a network component A sends a message to component B and B sends back an acknowledgement to A, it may be of interest to reason about whether B *knows* that A *knows* that B *knows* the message. We do of course not mean that the car wash machine, the web browser or the network components have consciousness, neither do we necessarily assume properties of knowledge from epistemology such as justification, truth, etc.[1] The metaphorical use of "know" usually means "have information", but the precise meaning will always be defined in a formal system such as the ones presented in this thesis. As always, the use of metaphors comes with a price: the danger of identifying the two concepts.

To be able to do a formal analysis, the properties of knowledge must be defined. To this end, *epistemic logic* is used.

1.1.1 Epistemic Logic

While there are many different proposals about how to model the logic of knowledge in the literature, the most popular are based on *modal epistemic logic*, henceforth called *epistemic logic* when no confusion can occur, with language and semantics (Hintikka, 1962) taken from modal logic. The book "Reasoning about Knowledge" by Fagin *et al.* (1995) is the standard reference in the field.

[1]In the literature, the metaphor "belief" is sometimes used instead of "knowledge", and "knowledge" used as true belief. I do not make this distinction.

Modal epistemic logic is based on classical propositional logic. The modal operator K is used to represent the fact that something is known. If p is the proposition that it rains in Bergen, then Kp is the proposition that it is known that it rains in Bergen. In systems of several agents, *multi-agent systems*, (i) the individual agents often have to reason about the knowledge held by the other agents, and (ii) we (the analysts or designers) need to reason about the individual knowledge held by the agents. For multi-agent systems, the standard modal logic is extended to a multi-modal logic by introducing a knowledge operator K_i for each agent i. These logics can be used to express propositions like $K_1K_2\neg K_1\neg p$ — agent 1 knows that agent 2 knows that agent 1 considers it possible that it rains in Bergen.

Examples of potential properties of knowledge are whether K_ip can be true without K_iK_ip being true, or K_ip and $K_i\neg p$ being true at the same time. Rather than being one logic with fixed properties of knowledge, epistemic logic is a collection of logics with different properties.

1.2 Finite Syntactic Epistemic States

Dennett (1987) suggests that many systems can be analyzed by *ascribing* attitudes such as knowledge, intentions, etc., to agents; this is called the intentional stance. Examples of such ascriptions of knowledge were mentioned earlier.

In this thesis I will, however, model the knowledge of *deliberative agents*. A deliberative agent acts according to knowledge obtained by observing and reasoning about its environment and represented symbolically, in contrast to a *reactive* agent who reacts on stimuli from the environment without symbolical deliberation. Of course, "knowledge" is used as a metaphor also in this context.

In principle, symbolical knowledge can be represented internally by different agents in different ways; e.g. in different types of data structures.

In this thesis I consider agents who represent their knowledge *syntactically*; as strings of symbols representing formulae in some logical language[2]. It is assumed that agents have a *storage* for such syntactical representations. For example, they may write them down on a piece of paper or store them in a database. The storage is not required to be permanent; it is possible to remove items from it between a point in time and the next. However, at a given point in time, each agent is assumed to have a fixed set of such syntactical objects — called a *syntactic epistemic state* (henceforth sometimes just "epistemic state").

Obviously, no agents can have a syntactic epistemic state which is not *finite*.

The goal in this thesis is an epistemic logic for agents with finite syntactic epistemic states.

1.3 The Logical Omniscience Problem

Modal epistemic logic have been successfully applied to puzzles involving complicated nested knowledge, such as the *Three Wise Men* puzzle. One of the reasons that modal epistemic logic lends itself to these kinds of puzzles, is that

[2]Konolige (1986a) calls this concept "sentential", and uses the term "syntactic" for first-order approaches to epistemic logic.

it is assumed in the puzzles that the agents which are modeled are *extremely intelligent*. One property of knowledge shared by all the variants of modal epistemic logic is that the agents know all the consequences of their knowledge, including all tautologies. This is, however, a property not normally associated with the concept "knowledge". For example, if someone knows the rules of chess, he will not automatically know whether white has a winning strategy. This problem is called *the logical omniscience problem (LOP)*. The real concept of knowledge contains facts held *explicitly*, rather than *implicitly*. Levesque (1984a) identifies these two different concepts of knowledge; *implicit* and *explicit* knowledge. Explicit knowledge is the *actual* knowledge held explicitly by an agent, which he can act upon and answer questions about; implicit knowledge is that which follow logically from explicit knowledge. Modal epistemic logic can be seen as a theory of implicit knowledge of real agents or, alternatively, as a theory of explicit knowledge for ideal agents with unlimited reasoning abilities. It fails, however, as a general theory of the (explicit) knowledge of real agents. Particularly, it fails as a theory of the explicit knowledge of agents with finite syntactic epistemic states. "Solving" the LOP by interpreting "knowledge" as implicit knowledge of course only changes the problem to finding a logic that describes explicit knowledge. This thesis presents a logic of explicit knowledge for agents with finite explicit epistemic states, without logical omniscience.

1.3.1 Logically Non-omniscient Agents

Some of the explanations of the fact that real agents lack logical omniscience found in the literature are that an agent lacks *awareness* of certain formulae, that it does not *focus on all issues simultaneously*, that it has an *incomplete reasoning mechanism*, that is has *bounded resources*, that it has different *states of mind* at different times, that it reasons in a *non-standard logic*, that it is restricted to computing solutions to problems of a given *complexity class*, that it fails to take all *relevant knowledge* into consideration when considering a possible conclusion, and others. Fagin & Halpern (1988, p. 40) suggest that the LOP "stems from a number of different sources", and list the first four of the sources just mentioned.

Aside from incomplete reasoning mechanism, each of the other explanations seems to be a special case of *bounded resources*, and it is not difficult to find examples of real agents who do not match each of the descriptions. (Mental) *action* is needed to reason, and real agents cannot do an unlimited number of actions, because they have limited resources. Although an incomplete reasoning mechanism can in theory be a source of logical non-omniscience orthogonal to bounded resources, in practice the latter is much more plausible. For example, the reason that a chess player does not know whether white has a winning strategy is that he cannot reason fast enough, or, equivalently, that he does not have enough time, and not that he does not know the rules of chess.

Thus, in a realistic model of deliberative agents, *lack of logical omniscience should be modeled by bounded resources rather than by logical ignorance*. Hintikka's semantics does not take bounded resources into account. Resources here are typically time and memory. In this thesis I make the simplifying assumption that the only bound on memory is that it must be finite. This is done in order to focus on the other bounded resource: time.

Examples such as the chess example motivate explanations such as inability to compute solutions to problems in certain complexity classes as mentioned above. However, in principle there is not a qualitative difference in doing simple and complex deductive reasoning, in e.g. using modus ponens one time or a million times. Not even simple consequences of knowledge is obtained automatically, but presupposes mental action. One cannot say that an agent *must* at a particular time know *anything* additional because it follows from his other knowledge, because he may not have computed it yet. A proper logic of explicit knowledge must make the distinction between knowledge and reasoning, and modal epistemic logic does not do that. This motivates the following description of explicit knowledge: *explicit knowledge has no closure*.

Of course, a completely "stupid" agent, unable to draw *any* consequences of its knowledge, is perfectly non-omniscient. Thus non-omniscience is only one part of a proper logic. A crucial property of a logic is that it allows modeling non-omniscient agents that nevertheless are intelligent. Non-omniscience and non-ignorance are not contradictory when the concept of knowledge is distinguished from the concept of reasoning; note the difference between the fact that, in a logical system, agent a's knowledge of a set of formulas Φ logically implies that agent a may know some other formula ϕ after using some resources and the fact that agent a's knowledge of Φ implies agent a's knowledge of ϕ. In other words, agents must – in principle – be able to find out any consequence of their knowledge, given enough time. For example, unlike a closure condition such as

> If the agent knows both p and $p \to q$ then he must also know q.

the following condition describes explicit knowledge properly[3]:

> If the agent knows both p and $p \to q$ and he has a mechanism that can deduce q from p and $p \to q$, then he will know q in the future if he chooses to.

Each use of "know(s)" in this latter condition means "know(s) explicitly", and the condition is a relation between the (explicit) knowledge at different points in time and reasoning.

1.4 A Logic of Finite Syntactic Epistemic States

The solution to the seemingly contradictory requirements that the agents should not be logically omniscient and be able to find out any consequences of their knowledge, is that two modalities are required: *knowledge and time*.

A logic of explicit knowledge will then be composed of three parts. First, a model of the concept of static explicit knowledge at a point in time. Second, a model of time. Third, a model of interaction between the two modalities: how explicit knowledge evolves dynamically over time as a result of reasoning, observation, communication, etc.

This thesis presents such a logic, and the three parts are introduced next.

[3]This description of explicit knowledge as presupposing action is the view taken by Duc (1997b), where an action corresponds to the selection of an inference rule in a model similar to the one in this thesis (discussed further in Section 9.6).

1.4.1 A Model of Static Explicit Knowledge

The model of the static explicit knowledge of agents with finite syntactic epistemic states (as described in Section 1.2) describes each agent as a finite set of formulae in a specified *object language* in which the agents can represent their knowledge about the world and about other agents' knowledge. There are neither closure or consistency conditions on epistemic states; an agent can know, at a given point in time, a contradiction and he can know ϕ without knowing $\neg\neg\phi$. The only condition on the epistemic states is that they must be finite.

An agent explicitly knows a formula at a point in time if and only if the formula is included in his epistemic state.

1.4.2 A Model of Time

Time is viewed as a sequence of discreet time points. A main difference between logical models of time is whether they are *linear* or *branching*. The latter allow reasoning about possible futures, and the most popular logic of this kind is *Computational Tree Logic (CTL)*. Branching time is selected as a model of time in this thesis, because it allows such reasoning about explicit knowledge as "the agent may know the formula in the next state" or "it is possible that the agent will never know the formula".

Recently, *Alternating-time Temporal Logic (ATL)* has been proposed as a generalization of CTL. ATL allows reasoning about *cooperation*, for example about facts such that "agent a and b can cooperate to make the formula true in the next state". As this kind of cooperation is very useful to explain how explicit knowledge can evolve in a multi-agent system, ATL is selected as a framework for modeling time.

1.4.3 A Model of Dynamic Explicit Knowledge

In addition to a symbolical storage, the agents described by the logic are equipped with *mechanisms*. A mechanism is a general model of reasoning; an agent can use the mechanism to obtain a new epistemic state from a current one. In the logic, we can reason about mechanisms in terms of *knowledge of rules*, as follows:

> If the agent knows both p and $p \to q$ and he knows modus ponens, then he can get to know q in the next state if he chooses to.

Very few restrictions are placed on mechanisms; they are neither required to be sound or complete.

Reasoning is not the only way knowledge can evolve. In general knowledge evolves through observation, and in a multi-agent system through communication. In this thesis I focus on communication; other types of observation can be modeled as communication from an "environment agent". Communication can be viewed as a generalization of reasoning; the latter can be seen as intra-agent communication. Thus, mechanisms are generalized to model also inter-agent communication.

1.5 Overview of the Thesis

The next preliminary chapter reviews background and motivation for the rest of the thesis. Review of related work is mostly postponed to Part IV. First, modal epistemic logic is formally presented. This presentation also serves as an introduction to logical concepts and conventions used throughout the thesis. The logical omniscience problem is then discussed in greater detail, particularly the source of it in modal epistemic logic including some approaches to alleviate the problem in order to model explicit knowledge. The last background section is on alternating-time temporal logic.

The main parts of the thesis are II and III, which present the models introduced in Sections 1.4.1 and 1.4.3. Although the latter is an extension of the former, the former is a proper and interesting logic on its own and not just a preliminary step towards the latter.

Part II presents a model of static finite syntactic epistemic states. First, in Chapter 3, the assumption about finite states is not yet made. A logical language for reasoning about syntactic states is introduced, and a semantics where states can possibly be infinite defined. This semantics will be useful when considering a finite semantics later. Chapter 4 presents a sound and complete logical system. In Chapter 5 the semantical restriction to finite states is made, together with soundness and completeness results. Extensions of the logical system with additional axioms, particularly axioms describing epistemic properties, and corresponding semantical changes, are discussed in Chapter 6. Axioms from modal logic discussed in Ch. 2 are used as examples.

Part III extends the model from Part II into a model of dynamic finite syntactic epistemic states. This model is not developed in as great detail as the model in Part II, but the model itself is more complex. Chapter 7 presents the model with a logical language for reasoning about how knowledge change over time as a result of reasoning and communication. The relation to ATL, upon which the model is based, is discussed. In Chapter 8 the well known Byzantine Generals problem is used to illustrate some of the aspects of the model; particularly communication.

Finally, the work is discussed in Part IV. Chapter 9 compares to selected earlier work; the model in Part II to previous models of syntactic representation, and the model in III to previous models of evolving knowledge. Particularly, the latter is compared to Alternating-time Temporal Epistemic Logic, another integration of epistemic logic and ATL. Chapter 10 provides a summary, conclusions and discussion of future work.

Chapter 2

Background

This chapter gives a brief overview of background material which is used in the following Parts II and III of the thesis. A complete overview of the field is not presented here; earlier work is presented and compared with the work in the thesis in Part IV.

Traditional modal epistemic logics are presented first. The discussion also introduces some general logical notation and concepts used in later chapters. The logical omniscience problem and its source in modal epistemic logic is discussed in Section 2.2. In Section 2.3, alternating time temporal logic is presented.

2.1 Modal Epistemic Logic

This section gives a quick overview over standard modal epistemic logic (by "standard" I mean an epistemic modal logic that is propositional (not quantified), monotonic and multi-modal).

There are (at least) two ways of characterizing modal logics. Here, these – and the relation between them – are discussed. The first is Hilbert-style proof theory; a syntactic approach. The second is model theory; a semantic approach. First, the syntax of well-formed epistemic formulae is defined.

Let Φ be a set of primitive propositions. The language of epistemic logic for a set of n agents, named $1, \ldots, n$, is the set of well-formed formulae (wffs) $\mathcal{L}_n(\Phi)$. The set of wffs is defined inductively:

- $\Phi \subseteq \mathcal{L}_n(\Phi)$
- If $\phi \in \mathcal{L}_n(\Phi)$, then $\neg\phi \in \mathcal{L}_n(\Phi)$
- If $\phi, \psi \in \mathcal{L}_n(\Phi)$, then $(\phi \wedge \psi) \in \mathcal{L}_n(\Phi)$
- If $\phi \in \mathcal{L}_n(\Phi)$, then $K_i\phi \in \mathcal{L}_n(\Phi)$ for $1 \leq i \leq n$

As usual, the use of the connectives $\vee$, $\rightarrow$ and $\leftrightarrow$ is allowed as syntactic sugar, and parentheses can be omitted when no confusion can occur.

A *logical system S* in $\mathcal{L}_n(\Phi)$ is a pair $(\mathcal{R}, \mathcal{A})$ where $\mathcal{A} \subseteq \mathcal{L}_n(\Phi)$ is a set of *axioms* and $\mathcal{R}$ is a set of (transformation) *rules*. For epistemic logic, $\mathcal{R}$ is the following set of rules:

$$\frac{\vdash_S \phi, \vdash_S \phi \to \psi}{\vdash_S \psi} \qquad \textbf{MP}$$

$$\frac{\vdash_S \phi}{\vdash_S K_i\phi}, 1 \leq i \leq n \qquad \textbf{Nec}$$

The rules are called modus ponens and the necessitation rule , respectively.

An important property of a logical system is that of its *theorems*. The fact that a wff ϕ is a theorem of the system S is denoted $\vdash_S \phi$. The set of theorems for a logical system is defined as follows. First, the axioms are theorems. Second, if $\frac{\vdash_S \phi_i, \dots, \vdash_S \phi_k}{\vdash_S \psi}$ is an *instance* of a rule – that is, a rule with wffs substituted for the letters $\phi, \psi, \dots$ – and all ϕ_i, $1 \leq i \leq k$, are theorems, then ψ is a theorem too. Note that a rule is a *schema*, describing the forms of the wffs it matches. For instance, **MP** says that if wffs ϕ and $\phi \to \psi$ are theorems then so is ψ for all wffs ϕ and ψ. A rule is a means for obtaining a new theorem from the set of axioms and/or previously obtained theorems. If a theorem ϕ is obtained by iteratively applying the rules in a logical system S in this manner, the sequence of axioms and new wffs obtained by rule application is called a (Hilbert-style) *proof*.

For epistemic logic the set $\mathcal{A}$ of *axioms* includes the axiom schema PC:

ϕ, where ϕ is a substitution instance of a propositional tautology **PC**

In addition, wffs commonly included as axioms are described by the following schemata[1], for $1 \leq i \leq n$

$$K_i(\phi \to \psi) \to (K_i\phi \to K_i\psi) \qquad \mathbf{K_i}$$
$$K_i\phi \to \neg K_i \neg \phi \qquad \mathbf{D_i}$$
$$K_i\phi \to \phi \qquad \mathbf{T_i}$$
$$K_i\phi \to K_i K_i \phi \qquad \mathbf{4_i}$$
$$\neg K_i\phi \to K_i \neg K_i \phi \qquad \mathbf{5_i}$$
$$\neg \phi \to K_i \neg K_i \phi \qquad \mathbf{B_i}$$

Which axioms should be included in a logical system is a philosophical question. Here, only *normal* systems are discussed, i.e. systems defining logics in which the $\mathbf{K_i}$ formulae are theorems. The logical system $(\mathcal{R}, \{\mathbf{PC}\} \cup \{\mathbf{K_i} : 1 \leq i \leq n\})$ is called K_n. Systems describing logics of belief rather than knowledge (doxactic logics) often include the $\mathbf{D_i}$ axioms instead of the $\mathbf{T_i}$ axioms commonly included in epistemic logics. The $\mathbf{4_i}$ and $\mathbf{5_i}$ axioms are called the positive and negative introspection axioms, respectively. Commonly used logical systems and their axioms (in addition to **PC**) are T_n with the $\mathbf{K_i}$ and $\mathbf{T_i}$

[1]In another tradition, an finite set of wffs is used instead of axiom schemata, in addition to the *uniform substitution* rule

$$\frac{\vdash_S \phi}{\vdash_S \phi[\psi_1/p_1, \dots, \psi_k/p_k]} \qquad \textbf{USub}$$

where $\phi[\psi_1/p_1, \dots, \psi_k/p_k]$ denotes the formula resulting from uniformly replacing the primitive propositions p_i with wffs ψ_i in the formula ϕ.

axioms, $S4_n$ with the $\mathbf{K_i}$, $\mathbf{T_i}$ and $\mathbf{4_i}$ axioms, $S5_n$ with the $\mathbf{K_i}$, $\mathbf{T_i}$, $\mathbf{4_i}$ and $\mathbf{5_i}$ axioms, $KD45_n$ with the $\mathbf{K_i}$, $\mathbf{D_i}$, $\mathbf{4_i}$ and $\mathbf{5_i}$ axioms and KTB_n with the $\mathbf{K_i}$, $\mathbf{T_i}$ and $\mathbf{B_i}$ axioms. One relationship between these logical systems, viz. the inclusion of theorems, can be shown graphically (Fig. 2.1). In the figure, $S \longrightarrow S'$ indicates that all the theorems of system S' are also theorems of system S.

$$\begin{tikzcd}
 & S5_n \arrow[dl] \arrow[dr] & \\
KTB_n \arrow[dr] & & S4_n \arrow[dl] \\
 & T_n \arrow[d] & \\
 & D_n \arrow[d] & KD45_n \arrow[l] \\
 & K_n &
\end{tikzcd}$$

Figure 2.1: Logical systems

Semantics is used to assert the truth-value of a formula. Hintikka (1962) based the semantics of epistemic formulae on the notion of *possible worlds*. The basic assumption of the possible worlds approach is that we can use a set of alternative possibilities, or alternative worlds, to reason about how the world may be. Each possible world is a model of propositional logic. Hintikka then defines an agent's *knowledge* as the agent's ability to tell the correct state of affairs, i.e. to discern between the alternative worlds. If an agent is unable to discern between two alternative worlds in which some particular state of affairs, say the time of the day, differs, the agent can be said to lack knowledge of this state of affairs. In each alternative world, an agent is viewed as considering a subset of the set of alternative worlds *possible*, and is said to know a fact if this fact is true in all the worlds considered possible. Note that in a world an agent considers possible, another agent might consider a set of alternative worlds possible, which gives meaning to propositions with nested knowledge such as K_1K_2p (in each world agent 1 considers possible, agent 2 considers possible a set of worlds where p is true in every world). Possible worlds semantics is formalized by *Kripke structures*. A *frame* F for n agents is a tuple $(S, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, where $\mathcal{K}_i$, $1 \leq i \leq n$, are binary relations, called *possibility relations*, over the set of possible worlds, or *states*, S. A Kripke structure M over the frame $(S, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ and the set of primitive propositions Φ is a tuple $(S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, where π – called an *interpretation* – is a truth assignment to the primitive propositions in each state: $\pi(s) : \Phi \rightarrow \{\mathbf{true}, \mathbf{false}\}$ for all $s \in S$. The possibility relation K_i identifies the worlds agent i considers possible relative to any given (actual) world, and the interpretation π states which basic facts are true in each alternative world. The fact that a wff ϕ is true in a state s in a structure $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ over Φ is denoted $(M, s) \models \phi$.

The $\models$ relation is defined by structural induction on the formula:

$$\begin{aligned}
(M,s) &\models p &&\Leftrightarrow && \pi(s)(p) = \textbf{true}, \text{ where } p \in \Phi && (2.1)\\
(M,s) &\models \neg\phi &&\Leftrightarrow && (M,s) \not\models \phi && (2.2)\\
(M,s) &\models (\phi \wedge \psi) &&\Leftrightarrow && (M,s) \models \phi \text{ and } (M,s) \models \psi && (2.3)\\
(M,s) &\models K_i\phi &&\Leftrightarrow && (M,s') \models \phi \text{ for all } (s,s') \in \mathcal{K}_i && (2.4)
\end{aligned}$$

A wff ϕ is *valid* in a structure $M = (S, \pi, \mathcal{K}_1, \dots, \mathcal{K}_n)$, written $M \models \phi$, iff $(M,s) \models \phi$ for all $s \in S$. A wff is *valid*, written $\models \phi$, if it is valid in all structures. It is *valid in a frame F*, written $F \models \phi$, if it is valid in all structures based on F. Finally, a wff is *valid in a class of structures (frames)* $\mathcal{M}$ ($\mathcal{C}$), written $\mathcal{M} \models \phi$ ($\mathcal{C} \models \phi$), iff it is valid in all structures in $\mathcal{M}$ (frames in $\mathcal{C}$). Similar terminology apply to satisfiability. A wff ϕ is a *logical consequence* of a set of wffs Ψ (or Ψ *logically implies* ϕ) with respect to a class of structures $\mathcal{M}$, written $\Psi \models \phi$ (with $\mathcal{M}$ implicitly understood) iff, for all $M \in \mathcal{M}$, $(M,s) \models \phi$ whenever $(M,s) \models \psi$ for all $\psi \in \Psi$. A logical system S for $\mathcal{L}_n(\Phi)$ is *sound* with respect to a class of structures $\mathcal{M}$ (a class of frames $\mathcal{C}$) iff whenever $\vdash_S \phi$ then $\mathcal{M} \models \phi$ ($\mathcal{C} \models \phi$), for all $\phi \in \mathcal{L}_n(\Phi)$. It is *complete* with respect to $\mathcal{M}$ ($\mathcal{C}$) iff whenever $\mathcal{M} \models \phi$ ($\mathcal{C} \models \phi$) then $\vdash_S \phi$.

There are correspondences between certain axiom systems and validity. A logical system is *characterized* by a class of structures (frames) iff it is sound and complete with respect to that class. The following correspondences hold[2]: K_n is characterized by the class of all structures (frames), D_n is characterized by the class of all serial[3] structures (frames), T_n is characterized by the class of all reflexive structures (frames), $S4_n$ is characterized by the class of all reflexive and transitive structures (frames), $S5_n$ is characterized by the class of all reflexive, transitive and symmetric structures (frames), $KD45_n$ is characterized by the class of all eucledian[4], serial and transitive structures (frames), and KTB_n is characterized by the class of all reflexive and symmetric structures (frames).

The framework outlined above is a tool for reasoning about knowledge in multi-agent systems. The language of epistemic logic is quite expressive. For instance, if p is the proposition that it is raining, the statement "If it is raining, then Mary knows that John considers it possible that it is raining" can be expressed as the wff $p \rightarrow K_{Mary}\neg K_{John}\neg p$. However, the semantics defined in this framework leads to certain non-intuitive properties of knowledge. As mentioned in the introduction, one of these properties is logical omniscience.

2.2 The Logical Omniscience Problem

A property of modal epistemic logic is that the agents always know all consequences of their knowledge; in particular they know all tautologies. The assumption that an agent knows all the consequences of its knowledge seems to be too strong when analyzing many non-trivial situations.

[2]A property, such as reflexivity, is said to hold for a structure (frame) when it holds for all the possibility relations in that structure (frame).

[3]A binary relation R over A is called *serial* iff for all $a \in A$ there exists a $b \in A$ such that $(a,b) \in R$.

[4]A binary relation R is called *eucledian* iff when $(a,b) \in R$ and $(a,c) \in R$ then $(b,c) \in R$.

As an example, consider the following cryptography scenario (a similar example is found in (Halpern, Moses, & Vardi, 1994)): agent a sends an encoded version m_e of a message m to agent c, in order to keep m secret from agent b, over a channel known to be insecure. m_e can be decoded, to m, by using two large primes n_1 and n_2, and the product $n_1 \times n_2$ is publicly known. If we make the assumption, as in modal epistemic logic, that the agents know all consequences of their knowledge, reasoning about their knowledge would proceed as follows. If we assume that agent b knows the rules of arithmetic (not an unreasonable assumption), then it follows that b can deduce the values of n_1 and n_2 from the public product and hence that once it knows m_e it also knows m. Furthermore, this fact is also known to agent a. Agent a's view of the knowledge held by agents c and b regarding n_1 and n_2 is then identical if a assume that b knows m_e. Of course, it is not realistic that agent b *knows* n_1 and n_2 just because it knows $n_1 \times n_2$ and an algorithm to deduce the former from the latter. In fact, this is why public-key cryptography schemes are useful – all the information needed to find the secret key is available but the computational complexity of the corresponding problem is proved to be very high.

In both this example and the chess example mentioned in the introduction, modal epistemic logic fails to model any real (human or artificial) agent properly.

The fact that according to modal epistemic logics agents are logically omniscient, i.e. know all consequences of their knowledge, and that real agents are not, is called *the logical omniscience problem (LOP)*, and was first identified by Hintikka (1975). In 1986, Moore (1986) said that the LOP was "perhaps the most hotly contested issue in this field [of reasoning about knowledge in artificial intelligence]", and it still receives considerable attention. Selected approaches to the problem is reviewed in Section 2.2.2 and in Chapter 9. Other reviews of the LOP include (Sim, 1997) and (Moreno, 1998). See also (Fagin *et al.*, 1995, Chap. 9) for a discussion. The source of the LOP in modal epistemic logic is discussed in Section 2.2.2. Presently, several forms of the LOP are presented.

2.2.1 Forms of Logical Omniscience

Several formulations of the LOP exist in the literature, and several "weaker" problems are commonly discussed under the LOP label. The original definition by Hintikka (1975) is the most common, and also the strongest condition: an agent knows all the logical consequences[5] of its knowledge. Fagin *et al.* (1995) call this *full* logical omniscience. Several other forms of omniscience exist, some of which are often called logical omniscience. And, as shown below, many are indeed instances of full logical omniscience. The following conditions are related to logical omniscience (Fagin *et al.*, 1995):

- *Closure under logical consequence (full logical omniscience)*: If ψ is a logical consequence of the set Φ and an agent knows all the formulae in Φ, then he also knows ψ.

[5]Logical consequence, and thus logical omniscience, is relative to a given class of structures. All the forms of logical omniscience discussed here are implicitly assumed to be relative to such a class. Also, as discussed previously, considering validity with respect to certain classes of structures is equivalent to considering provability with respect to the corresponding logical systems.

- *Knowledge of all valid formulae*: If ϕ is valid, then any agent knows ϕ. A related problem is the problem of *irrelevant beliefs*.
- *Closure under logical implication*: If ψ is a logical consequence of ϕ and an agent knows ϕ, then the agent also knows ψ.
- *Closure under logical equivalence*: If ϕ and ψ are logically equivalent and an agent knows ϕ, then the agent also knows ψ.
- *Closure under material implication*: If an agent knows both ϕ and $\phi \rightarrow \psi$, then the agent also knows ψ.
- *Closure under valid implication*: If $\phi \rightarrow \psi$ is valid and an agent knows ϕ, then the agent also knows ψ.
- *Closure under conjunction*: If an agent knows both ϕ and ψ, it also knows $\phi \wedge \psi$.

It is easy to see that knowledge of all valid formulae, closure under logical implication and closure under logical equivalence are implied by full logical omniscience. If we assume the standard interpretations of material implication and conjunction, then also the three latter conditions follow from full logical omniscience.

Fagin *et al.* (1995, p. 311) point out that "Logical omniscience can be viewed as a certain closure property of an agent's knowledge [...]". From the discussion in Ch. 1 about separating the notions of knowledge and reasoning, it may be tempting to generalize the concept of logical omniscience to *any* closure condition on knowledge[6]. There are, however, circumstances where particular closure conditions are appropriate. It may be, for example, that an agent's reasoning mechanism *never* will produce $\neg\neg\phi$ without also producing ϕ. The agent's explicit knowledge will then be closed under this requirement, because it is impossible for the agent to know $\neg\neg\phi$ without also knowing ϕ. Note, however, that this argument only holds for *finite* closure conditions; the other direction of the example — knowing ϕ implies knowing $\neg\neg\phi$ — cannot be explained by one step of a reasoning mechanism since it would mean that the mechanism could produce infinitely many inferences simultaneously. This motivates the following definition.

- *Partial Logical Omniscience:* If an agent knows ϕ, then there are infinitely many ψ such that the agent knows ψ.

Closure under logical consequence, henceforth "full logical omniscience", is implied by partial logical omniscience, as are all the other conditions listed above except closure under material implication.

As discussed in Chapter 1, the concern in this thesis is a logic describing the explicit knowledge of deliberative agents who represent their knowledge syntactically. It is assumed that the agents have finite epistemic states, and thus cannot be partially logically omniscient.

[6]For example (in the notation of modal epistemic logic):

- If $K_i\phi$ is a logical consequence of a set Φ, and there is no finite subset $\Phi' \subseteq \Phi$ such that $(\wedge\Phi') \rightarrow K_i\phi$ is an instance of a propositional tautology.

2.2.2 The Source of Logical Omniscience in Epistemic Logic

In standard (modal) epistemic logic all agents are fully logical omniscient (with respect to all subsets of all structures). To see this, consider a state s in a structure M where $(M, s) \models K_i\phi$ for each ϕ in a set Φ, and let $\Phi \models \psi$. Then, for each $(s, s') \in \mathcal{K}_i$, $(M, s') \models \phi$ for all $\phi \in \Phi$ and so $(M, s') \models \psi$ by logical consequence. Thus, $(M, s) \models K_i\psi$. In fact, due to the standard definition of the propositional connectives, modal epistemic logic suffers from all the forms of logical omniscience listed in Section 2.2.1.

In search of the cause of the LOP in modal epistemic logic, it is tempting to investigate the properties of knowledge in each particular logical system; i.e., which of the formulae listed on page 10 are valid in the system. However, full logical omniscience was shown above for all systems without any restriction on the possibility relations (i.e., without any assumptions of axioms other than $\mathbf{K_i}$). Thus, the LOP goes deeper than a particular axiomatization. The cause of the problem lies in the way Hintikka defined the concept of knowledge in the possible worlds framework. The requirements that, first, an agent views a subset of worlds as possible, second, that every world is a model of propositional logic and, third, that the agent *must* know something because that is true in all these worlds, are clearly sources of the problem. Chomsky (1982, p. 91) states that possible worlds semantics "fails" in propositional attitude contexts. Vardi (1986) points out that the assumption of a set of possible worlds is not problematic, but this association of knowledge with a subset of these worlds goes a long way beyond this assumption.

More fundamental critique of the use of possible worlds has also been offered. Hadley (1988) argues that it is unreasonable to assume that an agent knows the function mapping atomic propositions to truth values in all possible worlds if the set of possible worlds is large. Wooldridge (1995b) argues that possible worlds are not useful in practice unless they are "grounded", i.e., given a concrete interpretation.

Several authors have suggested to weaken the assumption that all worlds are models of propositional logic in order to solve the LOP. Although the formalisms will not be used in the remainder of the thesis, three of these approaches are briefly introduced in the next subsection, in order to show that approaches with non-logical worlds are not suitable as models of explicit syntactic knowledge. In Chapter 9, earlier work more directly related to the model of syntactic knowledge presented in the thesis is discussed in more detail.

Non-logical Worlds

Impossible Possible Worlds Hintikka (1975) responds to the claim that his proposed (Hintikka, 1962) possible worlds analysis of knowledge and other propositional attitudes commits us to the assumption that an agent knows all the consequences of its knowledge. He argues as follows that no such commitment is inherent in the approach. First, an agent knows a fact if it is true in all the worlds it considers epistemically possible. Second, a realistic agent is not logically omniscient, that is, there exist ϕ and ψ such that ϕ logically implies ψ and the agent knows ϕ but does not know ψ. Third, ϕ logically implies ψ if $\phi \rightarrow \psi$ is true in every logically possible world. Hintikka argues that these three facts are not incompatible unless we make the assumption that every *epistemically* possible world is also *logically* possible. This usually implicit assumption, he claims, is the root of the logical omniscience problem. The reason for this is that only a logical omniscient agent would only consider logically possible worlds possible; an agent with limited reasoning abilities would need to hold more options open (even though they really are logically impossible). We can give up this assumption by introducing worlds that the agents can consider possible, even though they are not logically possible. Hintikka (1975) calls these worlds *impossible possible worlds*[7]. The following formal treatment is from Fagin *et al.* (1995).

An *impossible worlds structure* is a tuple $M = (S, W, \sigma, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, where $(S, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ is a Kripke frame, $W \subseteq S$ is called the *possible states*, and σ is a syntactic assignment with the following properties in the possible states $s \in W$:

$$\sigma(s)(\phi \wedge \psi) = \mathbf{true} \Leftrightarrow \sigma(s)(\phi) = \mathbf{true} \text{ and } \sigma(s)(\psi) = \mathbf{true}$$
$$\sigma(s)(\neg\phi) = \mathbf{true} \Leftrightarrow \sigma(s)(\phi) = \mathbf{false}$$
$$\sigma(s)(K_i\phi) = \mathbf{true} \Leftrightarrow \sigma(s')(\phi) = \mathbf{true} \text{ for all } s' \text{ such that } (s, s') \in K_i$$

The satisfaction relation is defined as

$$(M, s) \models \phi \qquad \Leftrightarrow \qquad \sigma(s)(\phi) = \mathbf{true}$$

σ behaves in a standard way in logically possible worlds, but in the impossible possible worlds $S - W$, e.g. both (or neither) ϕ and $\neg\phi$ can be true. Since impossible possible worlds are only epistemical rather than logical alternatives, logical implication and validity are therefore defined relative to the set of possible states only. Ψ logically implies ϕ if for all impossible worlds structures M and possible states $s \in W$ in M such that $(M, s) \models \psi$ for all $\psi \in \Psi$, $(M, s) \models \phi$. ϕ is valid if, for all M, $(M, s) \models \phi$ for all possible states $s \in W$ in M. For example, assume that an agent knows all $\psi \in \Psi$ and that Ψ logically implies ϕ. Then ϕ must be true in all the logically possible states. The agent may, however, also consider other states where ϕ is not true possible, in which case it does not know ϕ.

Nonstandard Propositional Logic In Kripke structures, each world is a model of propositional logic. Fagin, Halpern, & Vardi (1996) suggest using worlds which are models for their nonstandard propositional logic NPL. The idea is to weaken logical omniscience by weakening the logical abilities of the agents. Knowledge and logical consequence are defined in the usual way. A difference from the impossible possible worlds approach is that all worlds are models of the nonstandard logic rather than having disjoint sets of possible and impossible worlds.

Several nonstandard propositional logics appears in the literature. Some of the best known are intuitionistic logic (Heyting, 1956) and relevance logic (Anderson & Belnap, 1975); the latter to which NPL is closely related. The particular nonstandard approach taken in NPL is to allow ϕ and $\neg\phi$ to have independent truth values. The intuition behind this is to consider an agent equipped with two databases; one for true formulae and one for false formulae. This idea is captured semantically by associating an *adjunct state* $s*$ to each

[7]The notion of such worlds has appeared several times in the literature under varying names. One of the first appearances is in Cresswell (1970), under the name *non-classical worlds*.

state s, and to define that $\neg\phi$ holds at s if and only if ϕ does not hold at $s*$. The decoupling of the semantics of ϕ and $\neg\phi$ destroys the semantics of material implication, since e.g. the formula $\phi \rightarrow \phi$ can be seen as an abbreviation of $\phi \vee \neg\phi$. This formula is not valid under the nonstandard semantics. To express the intuition behind material implication properly, a new connective $\hookrightarrow$ called *strong implication* is introduced. $\phi \hookrightarrow \psi$ is defined to be true if ψ is true whenever ϕ is true.

A *nonstandard Kripke structure* M is a tuple $(S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n, *)$ where $(S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ is a Kripke structure and $*$ is a function $* : S \rightarrow S$ (we write $s*$ for $*(s)$) such that $s** = s$. The satisfaction relation is defined exactly as for Kripke structures (particularly, knowledge is defined as truth in all worlds considered possible), except for the negation clause and a new clause for strong implication :

$$(M, s) \models \neg\phi \quad \Leftrightarrow \quad (M, s*) \not\models \phi \tag{2.5}$$

$$(M, s) \models \phi \hookrightarrow \psi \quad \Leftrightarrow \quad (M, s) \not\models \phi \text{ or } (M, s) \models \psi \tag{2.6}$$

Fagin, Halpern, & Vardi (1996) show that the logical system $K^{\hookrightarrow}$ consisting of the axiom schema

$$(K_i\phi \wedge K_i(\phi \hookrightarrow \psi)) \hookrightarrow K_i\psi$$

and the inference rules

All sound inference rules of NPL

and

$$\frac{\vdash_{K^{\hookrightarrow}} \phi}{\vdash_{K^{\hookrightarrow}} K_i\phi}$$

is sound and complete with respect to validity in nonstandard Kripke structures.

As mentioned above, all worlds in a nonstandard structure are models of the nonstandard propositional logic. However if $s* = s$, the semantics of negation in nonstandard structures (eq. 2.5) coincides with that of Kripke structures (eq. 2.2 on page 12). The state s is then called a *standard state*. It is possible to view nonstandard structures from the perspective of impossible possible worlds rather than the nonstandard logic approach discussed here, by viewing standard worlds as possible worlds and nonstandard worlds as impossible worlds. A nonstandard structure $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n, *)$ induces an impossible worlds structure $M' = (S, W, \sigma, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ where W is the set of standard states in M and $\sigma(s)(\phi) = \textbf{true}$ if and only if $(M, s) \models \phi$. A formula is said to be *standard-state valid* if it is true at every standard world in every nonstandard structure.

Explicit and Implicit Belief Levesque (1984a)[8] identifies two different concepts of knowledge; namely *implicit* and *explicit* knowledge, and his approach has been extended by or inspired others (Lakemeyer, 1987; Fagin & Halpern,

[8]Later revised in Levesque (1984b).

1988). Explicit knowledge is the *actual* knowledge held explicitly by an agent; implicit knowledge is that which follow logically from explicit knowledge.

Levesque describes a logic for both implicit and explicit knowledge based on impossible worlds or, as he calls them, *situations*. A situation is a "partial" possible world; a world that supports the truth of some sentences, the falsity of some sentences, and possibly neither the truth or falsity of other sentences. Incoherent situations, supporting both the truth and falsity of some sentence, are also allowed.

The language Levesque considers is quite restricted. Particularly, nested knowledge is not allowed. Lakemeyer (1987) extends Levesque's framework with nested knowledge.

Formally, the language is defined in the usual way from a set of atomic propositions $\mathbf{\Phi}$ and is closed under the connectives $\vee$, $\wedge$, $\neg$, B and L. The intended meaning of the formulae $B\phi$ and $L\phi$ is that ϕ is explicit and implicit knowledge, respectively.

The semantics of the language is defined through a model structure

$$M = (\mathcal{S}, \mathcal{B}, \mathcal{T}, \mathcal{F})$$

where $\mathcal{S}$ is a set of situations, $\mathcal{B} \subseteq \mathcal{S}$, and $\mathcal{T}$ and $\mathcal{F}$ are functions from $\mathbf{\Phi}$ to the powerset of $\mathcal{S}$. Given a model structure M, the support relations $\models_T$ and $\models_F$ between $\mathcal{S}$ and the language are defined. Below, $\mathcal{W}(s)$ denotes the set of situations that, for all $p \in \mathbf{\Phi}$ are a) members of exactly one of $\mathcal{T}(p)$ and $\mathcal{F}(p)$, b) is a member of $\mathcal{T}(p)$ whenever s is and c) is a member of $\mathcal{F}(p)$ whenever s is. $s \models_T \phi$ ($s \models_F \phi$) means that s supports the truth (falsity) of ϕ:

$$\begin{array}{lcl}
s \models_T p & \Leftrightarrow & s \in \mathcal{T}(p) \text{ where } p \in \mathbf{\Phi} \\
s \models_F p & \Leftrightarrow & s \in \mathcal{F}(p) \text{ where } p \in \mathbf{\Phi} \\
s \models_T (\phi \vee \psi) & \Leftrightarrow & s \models_T \phi \text{ or } s \models_T \psi \\
s \models_F (\phi \vee \psi) & \Leftrightarrow & s \models_F \phi \text{ and } s \models_F \psi \\
s \models_T (\phi \wedge \psi) & \Leftrightarrow & s \models_T \phi \text{ and } s \models_T \psi \\
s \models_F (\phi \wedge \psi) & \Leftrightarrow & s \models_F \phi \text{ or } s \models_F \psi \\
s \models_T \neg\phi & \Leftrightarrow & s \models_F \phi \\
s \models_F \neg\phi & \Leftrightarrow & s \models_T \phi \\
s \models_T B\phi & \Leftrightarrow & s' \models_T \phi \text{ for every } s' \in \mathcal{B} \\
s \models_F B\phi & \Leftrightarrow & s \not\models_T B\phi \\
s \models_T L\phi & \Leftrightarrow & s' \models_T \phi \text{ for every } s' \in \mathcal{W}(\mathcal{B}) \\
s \models_F L\phi & \Leftrightarrow & s \not\models_T L\phi
\end{array}$$

Finally, validity is defined as follows:

$$\models \phi \quad \Leftrightarrow \quad s \models_T \phi \text{ for any model structure } (\mathcal{S}, \mathcal{B}, \mathcal{T}, \mathcal{F}) \text{ and any } s \in \mathcal{W}(\mathcal{S})$$

Levesque then defines an axiomatic system for the logic. It turns out that explicit knowledge can be characterized by the notion of entailment in relevance logic (Anderson & Belnap, 1975), and the corresponding axioms and rules can thus be "imported" directly from relevance logic. Some of them are[9]:

$$B(\phi \wedge \psi) \leftrightarrow B(\psi \wedge \phi)$$
$$B\neg(\phi \vee \psi) \leftrightarrow B(\neg\phi \wedge \neg\psi)$$
$$B\phi \wedge B\psi \leftrightarrow B(\phi \wedge \psi)$$
$$\frac{\vdash ((B\phi \vee B\psi) \rightarrow B\gamma)}{\vdash B(\phi \vee \psi) \rightarrow B\gamma}$$

Non-logical Worlds as Models of Syntactic Knowledge All the three approaches with non-logical worlds weaken the logical omniscience problem, but do not solve it. Indeed, the agents described in all three approaches are logically omniscient with respect to weaker logics; they are partially logically omniscient.

For example, in Levesque's concept of explicit knowledge if an agent knows p then he knows $p \wedge p$ and $p \wedge p \wedge p$ and so on. Clearly, for syntactical knowledge each of these formulae must be computed to be explicitly known. Similarly, adding impossible possible worlds just changes the logic in which the agents reason and in the new logic they are logically omniscient (Vardi, 1986), and in the approach with nonstandard Kripke structures the agents are logically omniscient with respect to NPL.

Thus, neither modal epistemic logic nor attempts at removing the source of the LOP within the possible worlds framework can be seen as proper descriptions of explicit syntactic knowledge.

2.3 Alternating-time Temporal Logic

Temporal logics differ in whether they consider the future[10] to be linear or branching. *Computational Tree Logic (CTL)* is a commonly used branching time temporal logic. CTL has temporal connectives consisting of two parts, a path quantifier and a tense (or state) quantifier. For example, the formula $A\mathcal{F}\phi$ expresses that along all paths there is a state where ϕ is true. $E\Box\phi$ expresses that there is a path where ϕ is globally true, i.e. true at all states. The semantics of CTL is defined by Kripke structures.

Alternating-time Temporal Logic (Alur, Henzinger, & Kupferman, 2002) is a generalization of CTL in which the path quantifiers A and E are generalized to a set of quantifiers $\langle\langle G \rangle\rangle$ where G is a group of agents, in order to allow the expression of *cooperation* and *strategies*. Informally, $\langle\langle G \rangle\rangle\mathcal{F}\phi$ means that G can cooperate to make ϕ true in the future, and similar for the other state quantifiers. For example

$$\langle\langle a, b \rangle\rangle\mathcal{F}p$$

means that agents a and b have a strategy to ensure that, no matter what the other agents in the system do, p will be eventually be true if a and b use the strategy.

Note that the CTL quantifiers A and E can be expressed as $\langle\langle \emptyset \rangle\rangle$ and $\langle\langle A' \rangle\rangle$, where A' is the set of all agents in the system, respectively.

[9]Note that the last rule was erroneously omitted from Levesque (1984a), but was included in Levesque (1984b).

[10]Or the past or both; here I will only be concerned with the future.

ATL will be used in Chapter 7 to model agents whose knowledge changes over time and who can cooperate through communication.

2.3.1 Concurrent Game Structures

The semantics of ATL is defined by concurrent game structures, a generalization of Kripke structures.

Definition 2.1 (Alur, Henzinger, & Kupferman, 2002) A concurrent game structure[11] is a tuple

$$(k, Q, \Pi, \pi', d, \delta)$$

where

- $k > 0$ is a natural number of *players*
- Q is a finite set of *states*
- Π is a finite set of *propositions*
- $\pi'(q) \subseteq \Pi$ for each $q \in Q$; *the labeling function*
- For each player $a \in \{1, \ldots, k\}$ and state $q \in Q$, $d_a(q) \geq 1$ is a natural number. The set $\{1, \ldots, d_a(q)\}$ is called the set of *moves* available to player a in q. $D(q) = \{1, \ldots, d_1(q)\} \times \cdots \times \{1, \ldots, d_n(q)\}$ is the set of *move vectors* in q.
- For each move vector $v \in D(q)$ in a state $q \in Q$, $\delta(q, v) \in Q$; the *transition function*. $\square$

Intuitively, $\delta(q, (j_1, \ldots, j_k))$ is the next state of the system if each player i chooses move j_i in state q.

Computations and Strategies

A *computation* λ is an finite sequence of states; $\lambda = q_0 q_1 \cdots$, where for each $j \geq 0$ there is a move vector $v \in D(q_j)$ such that $\delta(q_j, v) = q_{j+1}$. We use $\lambda[j]$ to denote the element in λ with index j (q_j); $\lambda[0, j]$ to denote the finite $j + 1$ element prefix of λ ($q_0 \cdots q_j$).

The set of all non-empty *finite* state sequences is denoted Q^+; $Q^+ = \{q_1 q_2 \cdots q_m : q_j (1 \leq j \leq k) \in Q, m > 0\}$. A *strategy* for player i is a function $f_i : Q^+ \to \mathbb{N}$ having $f_i(q_1 \cdots q_m) \leq d_i(q_m)$, mapping any finite prefix of a computation to a move for player i. A *strategy vector* $\vec{f}_G$ for a set of players G is a set of strategies, one for each agent, $\vec{f}_G = \{f_i : i \in G\}$. The set of all strategy vectors for agents G is denoted $Str(G)$. A strategy vector $\vec{f}_G$ for $G \subseteq \{1, \ldots, k\}$ induces a set of computations $out(q, \vec{f}_G)$ for a given state $q \in Q$, called the *outcomes* of $\vec{f}_G$ in q, as follows. A computation $\lambda \in out(q, \vec{f}_G)$ iff

[11]Note that several slightly different definitions of concurrent game structures has been proposed by the authors. Many secondary papers use the definition in (Alur, Henzinger, & Kupferman, 1999). It turns out that the small differences in the definitions can be significant in certain contexts, as discussed in Section 9.4. In Section 7.4.1, I present yet another slightly different definition.

1. $\lambda[0] = q$
2. $\forall_{j\geq 0}\exists v = (m_1, \ldots, m_k) \in D(\lambda[j])$
   (a) $\forall_{i\in G} m_i = f_i(\lambda[0, j])$
   (b) $\delta(\lambda[j], v) = \lambda[j+1]$

In Part III a model of agents' mechanism for reasoning and communication is represented. This model induces a concurrent game structure, which will allow us to reason about the dynamic aspects of the system in terms of e.g. computations and strategies.

2.3.2 ATL Syntax and Semantics

Given a finite set Π of propositions and a finite number k of players, let $\Sigma = \{1, \ldots, k\}$. The following constitutes all well-formed ATL formulae:

- If $p \in \Pi$, then p is an ATL formula
- If ϕ_1 and ϕ_2 are ATL formulae, then $\neg\phi_1$ and $\phi_1 \vee \phi_2$ are ATL formulae.
- If ϕ_1 and ϕ_2 are ATL formulae and $A \subseteq \Sigma$, then $\langle\langle A\rangle\rangle \bigcirc \phi_1$, $\langle\langle A\rangle\rangle\Box\phi_1$ and $\langle\langle A\rangle\rangle\phi_1\mathcal{U}\phi_2$ are ATL formulae

Thus, the ATL language has 3×2^k operators consisting of two parts: a quantifier $\langle\langle A\rangle\rangle$ over *paths* or *computations* and a quantifier $\bigcirc$ ("next"), $\Box$ ("globally") or $\mathcal{U}$ ("until") over *states* along the paths.

The usual derived propositional connectives are used. A derived operator is $\langle\langle A\rangle\rangle\mathcal{F}$ ($\mathcal{F}$ meaning "sometime in the future"); $\langle\langle A\rangle\rangle\mathcal{F}\phi$ stands for $\langle\langle A\rangle\rangle\mathbf{true}\mathcal{U}\phi$.

The semantics of ATL is defined as follows.

Definition 2.2 If $S = (k, Q, \Pi, \pi', d, \delta)$ is a concurrent game structure, $q \in Q$, $A \subseteq \Sigma = \{1, \ldots, k\}$ and ψ is an ATL formulae, the satisfaction of ψ by S and q, written $S, q \models \psi$ is defined by structural induction over ψ:

$$S,q \models p \quad \Leftrightarrow \quad p \in \pi'(q), \text{ where } p \in \Pi \tag{2.7}$$
$$S,q \models \neg\phi \quad \Leftrightarrow \quad S,q \not\models \phi \tag{2.8}$$
$$S,q \models \phi_1 \vee \phi_2 \quad \Leftrightarrow \quad S,q \models \phi_1 \text{ or } S,q \models \phi_2 \tag{2.9}$$
$$S,q \models \langle\langle A\rangle\rangle \bigcirc \phi \quad \Leftrightarrow \quad \exists_{\vec{f}_A\in Str(A)}\forall_{\lambda\in out(q,\vec{f}_A)} S, \lambda[1] \models \phi \tag{2.10}$$
$$S,q \models \langle\langle A\rangle\rangle\Box\phi \quad \Leftrightarrow \quad \exists_{\vec{f}_A\in Str(A)}\forall_{\lambda\in out(q,\vec{f}_A)}\forall_{j\geq 0} S, \lambda[j] \models \phi \tag{2.11}$$
$$S,q \models \langle\langle A\rangle\rangle\phi_1\mathcal{U}\phi_2 \quad \Leftrightarrow \tag{2.12}$$
$$\exists_{\vec{f}_A\in Str(A)}\forall_{\lambda\in out(q,\vec{f}_A)}\exists_{j\geq 0}(S, \lambda[j] \models \phi_2 \text{ and } \forall_{0\leq k<j} S, \lambda[k] \models \phi_1) \tag{2.13}$$

□

Intuitively, $\langle\langle A\rangle\rangle \bigcirc \phi$,$\langle\langle A\rangle\rangle\Box\phi$ and $\langle\langle A\rangle\rangle\mathcal{F}\phi$ mean that the agents A *can cooperate* to make ϕ true in the next state, all states in the future, and some state in the future, respectively. $\langle\langle A\rangle\rangle\phi_1\mathcal{U}\phi_2$ means that A can cooperate to make ϕ_2 true in some future state and ϕ_1 true in every state before that.

Note that the expression $\langle\langle\emptyset\rangle\rangle\Box\phi$ means that ϕ will *always* be true — no matter what any of the agents do.

A derived path quantifier $[\![A]\!]$ is sometimes used. $[\![A]\!] \bigcirc \phi$ stands for $\neg\langle\langle A\rangle\rangle \bigcirc \neg\phi$, $[\![A]\!]\Box\phi$ for $\neg\langle\langle A\rangle\rangle\mathcal{F}\neg\phi$ and $[\![A]\!]\mathcal{F}\phi$ for $\neg\langle\langle A\rangle\rangle\Box\neg\phi$, meaning that the agents in *A cannot avoid* ϕ in in the next state, all states in the future, and some state in the future, respectively.

The language defined in Part III is based on the ATL language.

2.3.3 Variations

Presenting ATL, Alur, Henzinger, & Kupferman (2002) also discuss several variations of the general framework: specializations (Moore synchronous game structures, turn-based synchronous game structures) and generalizations (game structures with fairness constraints, ATL with incomplete information). Fairness constraints will not be considered further here.

Moore synchronous game structures

A game structure $S = (k, Q, \Pi, \pi', d, \delta)$ is *Moore synchronous* if:

1. Q is of the form $Q_1 \times \cdots \times Q_k$
2. For each player a there is a function δ_a mapping a state $q \in Q$ and a move $j \in \{1, \ldots, d_a(q)\}$ to Q_a such that $\delta(q, (j_1, \ldots, j_k)) = (\delta_1(q, j_1), \ldots, \delta(q, j_k))$.

Intuitively, each player has his own local state, and a player's next local state is determined by the move the player chooses along with the current global state (the composition of all the local states).

Turn-based Synchronous Game Structures

A concurrent game structure is *turn-based synchronous* if for every $q \in Q$ there exists an a_q, $1 \leq a_q \leq k$, such that $d_a(q) = 1$ for every $a \neq a_q$.

An equivalent definition is the following: a turn-based synchronous game structure is a tuple $S = (k, Q, \Pi, \pi, \sigma, R)$ where $\sigma : Q \to \{1, \ldots, k\}$ maps a state to a player and $R \subseteq Q \times Q$ is a total transition relation. Intuitively, only one player can move in each state. In state q it is $\sigma(q)$'s *turn*, and he can choose a new state q' where $(q, q') \in Q$.

Turn-based Synchronous Game Structures with Incomplete Information

A *turn-based synchronous game structure with incomplete information* is a turn-based game structure $S = (k, Q, \Pi, \pi, \sigma, R)$ together with a set Π_a of *observable propositions* for each player $1 \leq a \leq k$, such that:

- For each $1 \leq a \leq k$, there is a $p_a \in \Pi$
- $p_a \in \pi(q)$ exactly when $\sigma(q) = a$
- Let $\pi_a(q) = \pi(q) \cap \Pi_a$ and $\pi_{\tilde{a}}(q) = \pi(q) \setminus \Pi_a$:
    1. If $\sigma(q) = a$ and $(q, q') \in R$ then $\pi_{\tilde{a}}(q) = \pi_{\tilde{a}}(q') \setminus \{p_{\sigma(q')}\}$
    2. If $\sigma(q_1) = \sigma(q_2) = a$, $\pi_a(q_1) = \pi_a(q_2)$ and $(q_1, q_1') \in R$ then $(q_2, q_2') \in R$ for all states q_2' such that $\pi_a(q_2') = \pi_a(q_1')$ and $\pi_{\tilde{a}}(q_2') \setminus \{p_{\sigma(q_2')}\} = \pi_{\tilde{a}}(q_2)$.

A player can observe when it is his turn. Condition 1 says that the player can only influence propositions he can observe (in addition to the propositions p_a which are influenced implicitly by the act of making a move). Condition 2 says that the moving player's choice is from upon the non-observable propositions.

Conditions 1 and 2 ensure that we can write the transition relation R as a vector of relations $\{R_a \subseteq V_a \times V_a : 1 \leq a \leq k\}$ where V_a, called the set of *a-views*, is $\wp(\Pi_a)$.

The notion of a strategy in turn-based synchronous game structures with incomplete information is also modified according to the two conditions above: 1) the strategy only specifies the truth-/falsehood of the moving player's observable propositions in the next state and 2) the strategy is a function of the history of the observable propositions. Formally, a strategy for player a is a function $f_a : V_a^+ \rightarrow V_a$ mapping each finite sequence of a-views to an a-view, with the restriction that $(v_m, f_a(v_0 \dots v_m)) \in R_a$.

Finally, the notion of a well-formed formula is restricted syntactically. The restriction is that for a formula $\langle\langle A \rangle\rangle \bigcirc \phi$ all the *involved propositions* of ϕ must be observable by each of the agents in A. Informally, the involved propositions of a formula is the set of propositions occurring in the formula together with the sets of observable propositions for all the player names occurring in the formula. The argument[12] is that it makes no sense that an agent could have a strategy to reach a state where a certain proposition is true if the player cannot observe that proposition, and, similarly, a player cannot determine if he has reached a state where another player has a certain strategy if the first player cannot observe all the propositions observable by the second player.

Incomplete information will be of interest in Part III, where agents take actions based on local epistemic states.

[12] I disagree. See Section 7.4.2.

Part II

A Static Logic of Finite Syntactic Epistemic States

Chapter 3

Language and Semantics

3.1 Introduction

It is widely accepted that there is a need for theories describing what agents actually or explicitly know and can act upon, as opposed to what they implicitly know as described by modal epistemic logic, i.e. what follows logically from their explicit knowledge. Levesque (1984b) and others (see Section 2.2.2) have proposed to formalize the rules governing explicit knowledge. The model of explicit knowledge in this Part II of the thesis is a different approach. First, instead of describing closure conditions on explicit knowledge, it is assumed that explicit knowledge *has* been obtained, and a logic for reasoning about *static* explicit knowledge in a group of agents is constructed. A framework for reasoning about static knowledge is useful for analyzing the knowledge in a group of agents at an instant in time (or a time span when no epistemic changes are made), for example between computing deductions. Second, it is assumed that the agents are deliberative reasoners who represent their knowledge syntactically.

The need for reasoning about explicit knowledge is illustrated by the cryptography example mentioned earlier: an agent a sends an encrypted version m_e of a secret message m to agent c through a public channel, it is possible to decipher the message using two (large) prime numbers n_1 and n_2, and the product $n = n_1 \times n_2$ is publicly known. If we use the "implicit" knowledge concept, the sentence

$$\text{“agent } b \text{ knows } m\text{”} \tag{3.1}$$

could be derived from the sentences

$$\text{“agent } b \text{ knows } m_e\text{”} \tag{3.2}$$

$$\text{“agent } b \text{ knows } n\text{”} \tag{3.3}$$

assuming agent b knows the rules of arithmetic, since the values of n_1 and n_2 follows logically from the value of n.

However, even if we use the "explicit" knowledge concept, the sentence

$$\text{“agent } b \text{ does not know } m\text{”} \tag{3.4}$$

does *not* follow logically from sentences (3.2) and (3.3). Information about what the agent explicitly knows, does not make us able to deduce what he (explicitly) does *not* know. For example, agent b could have gotten to know m even before the message was sent. But if we add the sentence

$$\text{“sentences (2) and (3) describe all that is known by } b\text{”} \tag{3.5}$$

then sentence (3.4) follows from (3.2), (3.3) and (3.5). The concept of *only knowing* has been suggested to capture “all an agent knows”, but most approaches are in the context of *implicit* knowledge. As illustrated by the example, using the “implicit” knowledge concept does not allow us to deduce sentence (3.4) from (3.2), (3.3) and (3.5), because m is implicitly included in what is “only known” since it follows logically from other known facts. Thus, a proper logic for reasoning about knowledge combines reasoning about explicit knowledge with the concept of only knowing.

In the current part of the thesis, a logic for static finite explicit knowledge is constructed. *How* the agents obtain their explicit knowledge is not discussed before in Part III. Neither is the relation of this knowledge to reality of concern here; an agent can e.g. know false facts or contradictions. In other words, nothing is assumed in general about the closure or consistency of explicit knowledge.

First, the notion of “only knowing” in the context of explicit syntactic knowledge is discussed, before the object language — the language in which the agents represent their knowledge internally — is formalized. A meta language, or logical language, for reasoning about syntactic epistemic states of agents is introduced in Sections 3.4 and 3.5. One of the main goals in this thesis is to model agents with *finite* epistemic states. To this end, it turns out, a more general model where the agents are not restricted to finite states is useful as an intermediate result. Therefore, a semantics where agents are not restricted to finite states is presented in Section 3.6. A sound and strongly complete logical system for this semantics is presented in Chapter 4. In Chapter 5 the semantics is restricted to finite states. The resulting logic is called *Static Syntactic Epistemic Logic (*SSEL*)*. A key concept, *finitary theories*, which in effect describe axiomatizable agents is introduced. As discussed in Section 2.1, different variations of modal epistemic logic can be obtained by adding axioms corresponding to epistemic properties. In Chapter 6 such extensions of SSEL are investigated, and examples of well known axioms are discussed.

3.2 Only Knowing

We want to capture the concept “all an agent explicitly knows”. Everything an agent *implicitly* knows may be possible to describe by one single formula α; the agent implicitly knows everything that is logically entailed by the formula. If the operator K_I denotes the concept of implicit knowledge, we can express the agent's knowledge by $K_I\alpha$. Everything an agent *explicitly* knows, however, cannot be described by a single formula (if it knows more than one formula), because there is no closure condition for explicit knowledge. If the operator K_E denotes explicit knowledge, the formulae $K_E\alpha$ and $K_E\beta$ say that the agent knows α and β, but does not say anything about e.g. whether the agent knows the formula $\gamma = (\alpha \wedge \beta)$. We therefore need to express knowledge about *sets* of formulae. The fact that the formulae $\alpha_1, \ldots, \alpha_k$ are all that is explicitly known by agent i will be described by the formula

$$\Diamond_i\{\alpha_1, \ldots, \alpha_k\}$$

When we consider static sets of explicit knowledge, there are a number of aspects we can reason about, for example "the agent knows more than X" or "the agent knows less than X". The fact that *at least* the formulae $\alpha_1, \ldots, \alpha_k$ are known by i is formalized by the formula

$$\triangle_i\{\alpha_1, \ldots, \alpha_k\}$$

This formula says that agent i knows each α_i, but it may know more. The formula

$$\triangledown_i\{\alpha_1, \ldots, \alpha_k\}$$

is used to express the fact that agent i knows *at most* $\alpha_1, \ldots, \alpha_k$, i.e. that all he knows is included in the set but he may know less. Evidently,

$$\Diamond_i X \Leftrightarrow \triangle_i X \wedge \triangledown_i X$$

3.3 Semantic Assumptions

It is assumed that there is a group of n deliberative agents who may posses "pieces of knowledge" about the world as well as about the knowledge of other agents, obtained through observation or deliberation. Initially, no assumption is made about the finiteness of the storage; this semantic assumption is added later (Ch. 5). Presently, the model of the "pieces of knowledge" is introduced. It is assumed that the agents have the ability to represent finite *sets* of formulae, so that the "pieces of knowledge" can contain the expressions about only knowing discussed above. Although the set of all "pieces of knowledge" is not defined as a formal logical language, I will henceforth abuse the terminology and refer to the elements of this set as *formulae* and call the set *the object language*. It is assumed that the representation is identical for all the agents. In general, no assumption about the combination of formulae in this set is made – i.e. no assumption about the consistency, or about any closure condition, of the underlying deliberation mechanism is made. I will, however, show how such assumptions can be modeled in Chapter 6. Informally, the semantics of the language which will be used to reason about the agents' knowledge describes the group of agents as a collection of points in the subset lattice of all formulae. Formal semantics is presented in Section 3.6. It is shown later that certain epistemic properties of the agents can be modeled by making the set of possible points a proper subset of all points.

The object language $OL(n, \Theta)$ — just OL when no confusion can occur — is defined over a given set Θ of primitive propositions, for a group of n agents. Θ is assumed to be countable (this assumption might be dropped which would, however, complicate some proofs).

Definition 3.1 ($OL(n, \Theta)$) $OL(n, \Theta)$ is the least set such that:

- $OL_0 = \Theta$

- If $X \in \wp^{fin}(OL_k)$ then $\left.\begin{array}{c}\triangle_i X \\ \triangledown_i X\end{array}\right\} \in OL_{k+1}$

  If $\alpha, \beta \in OL_k$ then $\left.\begin{array}{c}\neg\alpha \\ (\alpha \wedge \beta)\end{array}\right\} \in OL_{k+1}$

- $OL(n, \Theta) = \bigcup_{k=0}^{\infty} OL_k$ □

where $\wp^{fin}(S)$ denotes the set of finite subsets of S. I use the common propositional connectives $\vee, \rightarrow, \leftrightarrow$ as syntactic sugar with the usual meaning, and $\Diamond_i X$ to stand for $(\triangle_i X \wedge \triangledown_i X)$.

The nesting of the power set construct, i.e. its application at each stage of the construction of OL captures the possible nesting of epistemic operators in the agents' knowledge. Examples of formulae for agent i (pieces of knowledge which are held by agent i) are (if $p, q \in \Theta$ are primitive propositions): p (agent i knows p), $p \rightarrow q$ (agent i knows that p implies q), $\triangledown_j\{p\}$ (agent i knows that agent j knows at most p, and $(\triangle_j\{p \rightarrow q\} \wedge p) \rightarrow q$ (agent i knows that if agent j knows that p implies q and p is true, then q is true).

A language for reasoning about explicit knowledge over OL needs a representation of the elements in OL – which again require a representation of sets of such elements. In the following, the *agent language* and the *term language* are defined by mutual recursion. The latter is simply a notation for sets of the former, while the former includes such sets in its definition.

The term language is considered first.

3.4 The Subset Lattice Term Language

The term language allows us to construct terms corresponding to the *finite* sets of OL formulae. The following definition is parameterized by an arbitrary agent language L, which will be instantiated in the following Section 3.5. For the moment, think of it as (something similar to) OL.

Definition 3.2 ($TL(L)$) Given a language L, the set of terms $TL(L)$ is the least set such that

- If $\alpha_1, \ldots, \alpha_k \in L$ then $\underline{\{}\alpha_1\underline{,}\ldots\underline{,}\alpha_k\underline{\}} \in TL(L)$

- If $T, U \in TL(L)$ then $\left.\begin{array}{c}(T \sqcup U) \\ (T \sqcap U)\end{array}\right\} \in TL(L)$ □

Formally, the term language $TL(L)$ is defined over the alphabet

$$A_L \cup \{\underline{\{}, \underline{,}, \underline{\}}, (,), \sqcup, \sqcap\}$$

where A_L is the alphabet of L. A term of the form $\underline{\{}\alpha_1\underline{,}\ldots\underline{,}\alpha_k\underline{\}}$ will, of course, be used to denote the set $\{\alpha_1, \ldots, \alpha_k\}$. Often, to simplify the notation, I will write the term $\underline{\{}\alpha_1\underline{,}\ldots\underline{,}\alpha_k\underline{\}}$ simply as $\{\alpha_1, \ldots, \alpha_k\}$ (and the term $\underline{\{\}}$ as $\emptyset$). It will be clear from the context when the expression $\{\alpha_1, \ldots, \alpha_k\}$ is a term, and when it is a mathematical expression of a set. Such terms are called *basic*. In the following, $\alpha, \beta, \ldots$ will be used to denote formulae in the language L and $T, U, \ldots$ denote general terms.

The parameter L will be omitted when it is clear from context.

The logical meta language of *term formulae* is constructed from the following atomic expressions: if T, U are terms then $T \doteq U$ is a well formed formula, and the set of all such atoms is closed under the propositional connectives $\neg$ and $\wedge$. The usual abbreviations are used, in addition to $T \preceq U$ for $T \sqcup U \doteq U$. The following definitions provide interpretation of terms and validity[1] of term formulae.

Definition 3.3 (Interpretation of Terms) Given a language L together with an interpretation of the elements of L into a set S

$$[\_] : L \to S$$

the interpretation of terms $TL(L)$

$$[\_] : TL(L) \to \wp^{fin}(S)$$

is inductively defined as follows:

- $[\underline{\{}\alpha_1\underline{,}\ldots\underline{,}\alpha_n\underline{\}}] = \{[\alpha_1], \ldots, [\alpha_n]\}$
- $[T \sqcup U] = [T] \cup [U]$
- $[T \sqcap U] = [T] \cap [U]$ □

Validity of term formulae is defined as expected:

Definition 3.4 (Validity of Term Formulae)

$$\begin{array}{lcl} \models T \doteq U & \Leftrightarrow & [T] = [U] \\ \models \neg\phi & \Leftrightarrow & \not\models \phi \\ \models \phi \wedge \psi & \Leftrightarrow & \models \phi \text{ and } \models \psi \end{array}$$ □

Term formulae are a part of the meta language for the logic, which is introduced next.

3.5 The Epistemic Language

This section defines an *agent language* $AL(n, \Theta)$ for representing formulae in $OL(n, \Theta)$, and an *epistemic language* $EL(n, \Theta)$ which is a meta language for expressing propositions about sets of formulae in $OL(n, \Theta)$. These sets are represented by the term language $TL(AL)$. The epistemic language is a superset of the agent language and includes, in addition, term formulae over $TL(AL)$.

Definition 3.5 ($AL(n, \Theta)$) Given a set of primitive formulae Θ, the agent language $AL(n, \Theta)$, or just AL, is the least set such that:

- $\Theta \subseteq AL$
- If $T \in TL(AL)$ then $\left.\begin{array}{c} \triangle_i T \\ \triangledown_i T \end{array}\right\} \in AL$

[1]Truth of term formulae follows entirely from the particular formula and is not defined relative to an external structure; thus the notions of truth and validity coincide.

- If $\alpha, \beta \in AL$ then $\left.\begin{array}{c}\neg\alpha \\ (\alpha \wedge \beta)\end{array}\right\} \in AL$ $\square$

Definition 3.6 ($EL(n, \Theta)$) Given a set of primitive formulae Θ, the epistemic language $EL(n, \Theta)$, or just EL, is the least set such that:

- $AL \subseteq EL$
- If $T, U \in TL(AL)$ then $(T \doteq U) \in EL$
- If $\phi, \psi \in EL$ then $\left.\begin{array}{c}\neg\phi \\ (\phi \wedge \psi)\end{array}\right\} \in EL$ $\square$

As usual, the abbreviations $(\phi \vee \psi)$ for $\neg(\neg\phi \wedge \neg\psi)$, $(\phi \rightarrow \psi)$ for $(\neg\phi \vee \psi)$, $(\phi \leftrightarrow \psi)$ for $((\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi))$ and $T \preceq U$ for $T \sqcup U \doteq U$ are used. In addition, $\Diamond_i\phi$ stands for $(\triangle_i\phi \wedge \triangledown_i\phi)$. The operators $\triangle_i, \triangledown_i$ and $\Diamond_i$ are called epistemic operators. Paranthesis will sometimes be skipped when not necessary. The usual precedence of propositional connectives is used; e.g. $\phi_1 \wedge \phi_2 \rightarrow \phi_3 = ((\phi_1 \wedge \phi_2) \rightarrow \phi_3)$.

In the following, $p, q, \ldots$ are used to denote members of Θ, the meta-variables $\alpha, \beta, \ldots$ AL-formulae and $\phi, \psi, \ldots$ EL-formulae. The only instantiation of the parameter L in the definition of $TL(L)$ of concern in this Part II of the thesis is AL, and I will often write TL for $TL(AL)$.

The set $EL(n, \Theta)$ is defined by reference to $AL(n, \Theta)$ and this latter one by a mutual recursion with the definition of the set of terms $TL(AL)$ from Definition 3.2. Intuitively, and roughly:

$TL(AL)$ consists of terms denoting finite sets of AL formulae with arbitrary nesting, e.g., $\{p, q, r\}$, $\{p, \triangle_i\{p, q\}\}$, $\{p, \triangle_i\{p, \triangledown_j\{q\}\}$, $\{p, \triangle_i\{q\}\} \sqcup \{\triangledown_j\{p, q, r\}\}$. A term $T \in TL(AL)$ will be interpreted as a set $[T] \in OL$.

AL consists of formulae from Θ, applications of epistemic operators to terms, and their propositional combinations, e.g., $p, \triangle_i\{p, \triangledown_j q\}, \neg \triangle_i \{p\} \wedge \triangledown_i(\{p\} \sqcup \{q, r\})$. A formula in AL represents an element in OL.

EL extends AL with propositional combinations of term formulae $T \doteq U$, $T, U \in TL(AL)$.

The primitive term formulae are included in the epistemic language in order to increase the expressiveness and make it more convenient to write down, e.g., axiom schemata. The assumption about the agent language, however, is that no reasoning about sets are necessary. In Appendix A it is shown that these languages defined by mutual recursion are well-defined, by showing that the corresponding recursive function has a least fixed point.

3.6 Semantics

The agent language is intended to represent OL; generally, since we can now write down a set in several ways (using different orderings of the formulae and/or using set operators), more than one formula in the agent language represents a particular "piece of knowledge" in OL. The interpretation of AL-formulae as a function $[\_] : AL(n, \Theta) \rightarrow OL(n, \Theta)$ is now defined. This definition is henceforth assumed in the interpretation of $TL(AL)$ (Definition 3.3).

Definition 3.7 The interpretation of agent formulae $[\_] : AL(n,\Theta) \to OL(n,\Theta)$ is defined inductively:

- $[p] = p$ for $p \in \Theta$
- $[\neg\alpha] = \neg[\alpha]$, for $\alpha \in AL(n,\Theta)$
- $[\alpha_1 \wedge \alpha_2] = [\alpha_1] \wedge [\alpha_1]$, for $\alpha_1, \alpha_2 \in AL(n,\Theta)$
- $[\triangle_i T] = \triangle_i[T]$, for $T \in TL(AL)$
- $[\triangledown_i T] = \triangledown_i[T]$, for $T \in TL(AL)$

where the interpretation of terms $[\_] : TL(AL) \to \wp^{fin}(OL)$ is as in Definition 3.3. □

To determine the truth of formulae $\triangle_i T_1$ and $\triangledown_i T_2$, it must be determined whether the *OL*-formulae known by agent *i* includes $[T_1]$ or is included in $[T_2]$, respectively. Thus, a semantical structure must include information about the exact knowledge of the agents. Recall that the agents are not (yet) assumed to have finite knowledge. A naïve idea is then to represent the epistemic state of each agent *i* as a set $s_i \in \wp(OL)$ (in addition to a truth assignment to the primitive propositions Θ). Consider, however, the following theory:

$$\Gamma_1 = \{\triangle_b\{m_e\}, \neg \triangledown_b \{m_e\}\} \cup \{\neg \triangle_b \{\alpha\} : \alpha \neq m_e\}$$

Γ_1 describes a situation where

1. Agent *b* knows m_e
2. The set $\{m_e\}$ is not all which is known by agent *b*
3. Agent *b* does not know any other formula than m_e

Clearly, Γ_1 is not satisfiable in the naïve semantics, because there is no set of *OL*-formulae which describes agent *b*'s epistemic state[2]. Thus, Γ_1 should be considered inconsistent. However, there can be no finite proof of inconsistency of theories such as Γ_1, and an axiomatization would thus need a deduction rule with infinitely many antecedents.

An alternative to making Γ_1 inconsistent is making it satisfiable. The question is what Γ_1 could mean, if the assumptions of the naïve semantics are lifted. The problem with the naïve semantics is the correspondence between *AL* and *OL* — for each $\alpha \in OL$ there is an $\alpha' \in AL$ such that $[\alpha'] = \alpha$. If, on the contrary, there was an element $* \in OL$ such that $[\alpha'] \neq *$ for all $\alpha' \in AL$, then Γ_1 would be satisfied by a structure where the epistemic state of *b* is $\{m_e, *\}$. Thus, Γ_1 could be interpreted as describing an agent who knows something not expressible in the meta language in addition to m_e, instead of as a contradiction.

This idea is used in the following definition of a *general knowledge set structure* which represents the agents as a collection of points in the subset lattice of *OL* extended with $*$ (the reason $*$ is only included in *finite* subsets is discussed later) together with a truth assignment to the primitive propositions, and will be used to determine truth of the language *EL*.

[2]Suppose *b*'s epistemic state is $s_b \subseteq OL$. By point 1, $m_e \in s_b$. If $s_b = \{m_e\}$, then *b* knows at most m_e which contradicts point 2. If $\alpha \in s_b$ for $\alpha \neq m_e$, then it contradicts point 3.

Definition 3.8 (General Knowledge Set Structure) A General Knowledge Set Structure (GKSS) is an $n+1$-tuple

$$M = (s_1, \ldots, s_n, \pi)$$

where

$$s_i \in \wp(OL) \cup \wp^{fin}(OL \cup \{*\})$$

and $\pi : \Theta \to \{\mathbf{true}, \mathbf{false}\}$ is a truth assignment. s_i is the epistemic state of agent i, and the set of all epistemic states is $\mathcal{S} = \wp(OL) \cup \wp^{fin}(OL \cup \{*\})$. The set of all GKSSs is denoted $\mathcal{M}$. □

Any "piece" of knowledge an agent may have is finite. $OL(\Theta)$ (Def. 3.1) is the collection of such finite "pieces". An epistemic state can, however, have infinitely many (finite) pieces of knowledge. I will later be particularly interested in agents with only finite epistemic states, and will restrict the set of structures accordingly (Chapter 5). We may also want to model certain epistemic properties of the agents. Examples of this are an agent who always believes β whenever it believes both α and $(\alpha \to \beta)$, or an agent who never believes a contradiction. A result in Chapter 6 is that many interesting properties can be captured simply by restricting the set of allowed epistemic states for each agent.

We can view a GKSS as a description of the agents as points in the lattice $\mathcal{S}$. As an example, the point $\{\alpha, \triangle_k\{\alpha\}, \triangle_i\{\triangledown_j\{\alpha, \beta\}\}\}$ represents the epistemic state in which an agent (who is at this point) knows: 1) α, 2) that k knows α, and 3) that i knows that j knows at most α and β.

The point s_i represents all the formulae agent i knows. Notice that an agent may "know" formulae which are not true, just as he may "know" both α and $\neg\alpha$.

Definition 3.9 (Satisfaction) Satisfaction of a EL-formula ϕ in a GKSS $M = (s_1, \ldots, s_n, \pi) \in \mathcal{M}$, written $M \models \phi$, is defined as follows:

$$\begin{array}{lcl}
M \models p & \Leftrightarrow & \pi(p) = \mathbf{true} \\
M \models \neg\phi & \Leftrightarrow & M \not\models \phi \\
M \models (\phi \wedge \psi) & \Leftrightarrow & M \models \phi \text{ and } M \models \psi \\
M \models \triangle_i T & \Leftrightarrow & [T] \subseteq s_i \\
M \models \triangledown_i T & \Leftrightarrow & s_i \subseteq [T] \\
M \models T \doteq U & \Leftrightarrow & [T] = [U]
\end{array}$$
□

Notice an important and possibly confusing subtlety. A formula which is both an AL and an EL formula, say $\phi = \triangle_i\{p, q, \triangledown_j\{a, b\}\}$ is interpreted, according to Def. 3.7 as an element of OL. This, however, is merely an auxiliary definition needed for interpretation of terms. To check its satisfaction in a structure, we actually go to the point denoted by the term following the outermost epistemic operator – which is here the set $\{p, q, \triangledown_j\{a, b\}\}$ – and check whether the agent i is at or above this point. However, if this same formula ϕ occurred within a term in another formula, say $\psi = \triangledown_k\{\phi\}$ then, to check the satisfaction of ψ, we would only have to check whether k is at or below the point $\{\phi\}$.

The truth conditions for the derived operators are easily seen to be:

$$\begin{array}{lcl} M \models (\phi \vee \psi) & \Leftrightarrow & M \models \phi \text{ or } M \models \psi \\ M \models (\phi \rightarrow \psi) & \Leftrightarrow & M \not\models \phi \text{ or } M \models \psi \\ M \models \Diamond_i T & \Leftrightarrow & [T] = s_i \\ M \models T \preceq U & \Leftrightarrow & [T] \subseteq [U] \end{array}$$

An *EL*-formula ϕ is *valid* with respect to a class $\mathcal{M}' \subseteq \mathcal{M}$ of general knowledge set structures iff $M \models \phi$ for all $M \in \mathcal{M}'$. The model class of $\Gamma \subseteq EL$ is

$$mod(\Gamma) = \{M \in \mathcal{M} : M \models \gamma \text{ for all } \gamma \in \Gamma\}$$

A formula ϕ is a logical consequence of a set of formulae Γ, $\Gamma \models \phi$, iff ϕ is valid with respect to $mod(\Gamma)$.

Example 3.1 Consider again the encrypted message example (p. 27). If we assume that m_e, m, n are primitive propositions expressing the values of the variables in question, sentences (3.2), (3.3), (3.5) and (3.4) can be expressed as

$$\triangle_b\{m_e\} \tag{3.6}$$
$$\triangle_b\{n\} \tag{3.7}$$
$$\triangledown_b\{m_e, n\} \tag{3.8}$$
$$\neg \triangle_b \{m\} \tag{3.9}$$

It is easy to see that the latter is a logical consequence of the three former:

$$\{\triangle_b\{m_e\}, \triangle_b\{n\}, \triangledown_b\{m_e, n\}\} \models \neg \triangle_b \{m\}$$

If the case was that we did not know whether agent b have more information than the encrypted message and the product, we would remove formula 3.8. Now, formula 3.9 does not follow from 3.6 and 3.7. □

EL extends the language of term formulae, and the semantics of the latter part of *EL* is still independent on any structure (Def. 3.4): if ϕ is a term formulae then $M \models \phi \Leftrightarrow \models \phi$ for any M, and $\not\models \phi \Rightarrow \models \neg\phi$.

3.6.1 The Element $*$

The special element $*$ is used in the semantic description of an epistemic state syntactically described by a theory (i.e. a set of formulae) Γ of a special class of infinite theories – i.e. those in which

a) an agent cannot be at or below any (finite) point corresponding to a term ($\Gamma \cup \{\triangledown_i T\}$ is unsatisfiable for every T) and

b) the set of terms describing finite points the agent can be *above* is finite (there exists a "largest" term S such that $\Gamma \cup \{\triangle_i S\}$ is satisfiable and if $\Gamma \cup \{\triangle_i S'\}$ is satisfiable then $[S'] \subseteq [S]$)

In other words, the agent must be at a finite point which has no corresponding term. A state s where $* \in s$ represents such a point.

The formula $*$ which can be known but not expressed in the meta language is needed in the model, since we allow the expression of such situations in the language (by infinite theories).

One can think of states which include the element $*$ as infinite, because they are not bounded above by any term (point a above). Recall that the assumption that epistemic states can be infinite is only intermediate, and from Chapter 5 and through the rest of the thesis epistemic states are restricted to be finite. The $*$ element is then dispensed with. The results involving completeness with respect to GKSSs (with the $*$ element) in the next chapter will, however, be very useful later.

It is impossible to make the type of infinite theories such as Γ_1 on p. 33 inconsistent with finite deduction rules. The sound and complete logical system presented in the next chapter has only finite rules. As an alternative, we could dispense with $*$ and introduce infinite rules — a less desirable alternative[3].

3.6.2 On An Incompleteness of Knowledge

Here, one property of the introduced logic of syntactic epistemic states is shown. In the following theorem, the $\mathbf{T_i}$ axioms (knowledge is truth) discussed in Section 2.1 are adopted. In *EL* notation:

$$\triangle_i \{\alpha\} \to \alpha \qquad \mathbf{T_i}$$

Theorem 3.1 For any term S:

$$\mathbf{T_i} \models \neg \triangle_i \{\triangledown_i S\} \qquad \square$$

PROOF It is easy to see that $\mathbf{T_i}$ is satisfiable[4]. Let $M \models \mathbf{T_i}$, and assume that $M \models \triangle_i\{\triangledown_i S\}$ for some S, i.e. that $\triangledown_i[S] \in s_i$. By $\mathbf{T_i}$, $M \models \triangledown_i S$ and $s_i \subseteq [S]$. Then $\triangledown_i[S] \in [S]$, which is impossible. ■

Theorem 3.1 can be read as an epistemic analogue of Gödels incompleteness theorem: there exist true formulae which are impossible to know — more specifically, it is impossible to know that "this is all I know" (if knowledge is truth). Furthermore, this holds even for (imaginary) agents with infinite memory[5].

[3]Let

$$R_* = \frac{\text{For all } \alpha \in AL\text{: } \Gamma \vdash \triangle_i\{\alpha\} \to \{\alpha\} \preceq T}{\Gamma \vdash \triangledown_i T}$$

Let $\mathcal{M}^- = \{(s_1, \ldots, s_n, \pi) \in \mathcal{M} : * \notin s_i\}$. In Section 4.3 I describe a logical system *EC* which is proven sound and complete with respect to $\mathcal{M}$. To see that $EC \cup \{R_*\}$ is sound and complete with respect to $\mathcal{M}^-$, it then suffices to show that, for any maximal consistent theory Γ, if $\Gamma \vdash \neg \triangledown_i T$ for every T and $\bigcup_{\Gamma \vdash \triangle_i S}[S]$ is finite then Γ is inconsistent. Let $[S'] = \bigcup_{\Gamma \vdash \triangle_i S}[S]$, and let β be such that $\Gamma \vdash \neg\{\beta\} \preceq S'$. By consistency of Γ, $[\beta] \notin [S']$. By the definition of S', $\Gamma \nvdash \triangle_i\{\beta\}$ and by maximality $\Gamma \vdash \neg \triangle_i \{\beta\}$. By R_*, $\Gamma \vdash \triangledown_i S'$. Thus, since $\Gamma \vdash \neg \triangledown_i S'$ by assumption, Γ is inconsistent.

[4]Consider for example a structure with $s_i = \{p\}$ and $\pi(p) = \textit{true}$.

[5]A formula like $\triangle_i\{\triangledown_j S\}$, $i \neq j$, is of course fully consistent with $\mathbf{T_i}$; an agent can potentially know all *another* agent knows.

Example 3.2 Consider once again the encrypted message example from the introduction. We can express the fact that agent a knows that agent b knows at most m_e and n by the formula

$$\phi = \triangle_a\{\bigtriangledown_b\{m_e, n\}\}$$

It is easy to see that ϕ is satisfiable in $mod(\mathbf{T_i})$; let $\bigtriangledown_b\{m_e, n\} \in s_a$ and $s_b \subseteq \{m_e, n\}$.

The fact that agent b knows that it knows at most m_e and n, i.e.

$$\phi_1 = \triangle_b\{\bigtriangledown_b\{m_e, n\}\}$$

is clearly unsatisfiable in $mod(\mathbf{T_i})$ since the epistemic state of b must include the formula $\bigtriangledown_b\{m_e, n\}$ but in $mod(\mathbf{T_i})$ in every model of ϕ_1 the formula

$$\bigtriangledown_b\{m_e, n\}$$

must be true, i.e. the epistemic state of b can at most include m_e and n. If we add this formula to the description of the epistemic state of agent b, we get

$$\phi_2 = \triangle_b\{\bigtriangledown_b\{m_e, n, \bigtriangledown_b\{m_e, n\}\}\}$$

but in every model of ϕ_2 the formula

$$\bigtriangledown_b\{m_e, n, \bigtriangledown_b\{m_e, n\}\}$$

must be true, so ϕ_2 is also unsatisfiable. Continuing in the same way, we get

$$\phi_3 = \triangle_b\{\bigtriangledown_b\{m_e, n, \bigtriangledown_b\{m_e, n\}, \bigtriangledown_b\{m_e, n, \bigtriangledown_b\{m_e, n\}\}\}\}$$
$$\vdots$$

ad infinitum, and there is no ϕ_i which is satisfiable in $mod(\mathbf{T_i})$. □

The property (Th. 3.1) is a consequence of an impossibility of self-reference of terms: a (finite) term cannot be its own proper subterm.

Chapter 4

A Logical System

4.1 Introduction

In this chapter a logical system for the epistemic language *EL* is presented. In Section 2.1 several general logical concepts and definitions, such as *logical systems* and *proofs*, were presented. In addition, the following definitions are used.

A formula ϕ is provable in a system $S = (\mathcal{R}, \mathcal{A})$ *from a set of premises* Γ, written $\Gamma \vdash_S \phi$, iff it is a theorem of the system $S = (\mathcal{R}, \mathcal{A} \cup \Gamma)$. The notion of completeness introduced in in Section 2.1, $\models \phi \Rightarrow \vdash \phi$, can be called *weak* completeness. Recall also the notion of logical consequence form that section. In the GKSS semantics logical consequence, $\Gamma \models \phi$, can be defined as $M \models \Gamma \Rightarrow M \models \phi$ for every $M \in \mathcal{M}$. The notions of provability from premises and logical consequence leads to the concept of *strong* completeness: $\Gamma \models \phi \Rightarrow \Gamma \vdash \phi$. In the following, it will be clear from context which version of completeness is meant by the terms "complete" and "completeness".

An (*EL*) *theory* is a (possibly infinite) set of *EL*-formulae. A theory Γ is (S-) *inconsistent* iff both $\Gamma \vdash_S \phi$ and $\Gamma \vdash_S \neg\phi$ for some *EL*-formula ϕ, in which case $\Gamma \vdash_S \psi$ for *every* formula ψ if S is based on propositional logic. A theory Γ is *maximal* iff either $\Gamma \vdash_S \phi$ or $\Gamma \vdash \neg\phi$ for every formula ϕ. The subscript on $\vdash$ will be dropped when it is clear from context.

First, in Section 4.2, a term calculus for the part of *EL* called the term language is defined. This will allow us to concentrate in the following Section 4.3 exclusively on the epistemic operators. The relation between the developed calculus and the semantics for agents with possible infinite epistemic states from the previous chapter is shown by proving soundness and completeness. In the following chapter, the semantics are restricted to agents with finite epistemic states.

4.2 Term Calculus

In this section a *term calculus* is presented – a logical system for propositional combinations of term equations. It has two components. The first, the lattice calculus, is a generic axiomatization of a power set lattice. This is introduced in Section 4.2.1. The other component is an axiomatization of equality and inequality of basic terms which, again, involves interaction with the definition of the agent language. This is presented in Section 4.2.2. The term calculus is the union of these two components, and its soundness and completeness is shown in Section 4.2.3.

Recall from Section 3.4 that the language of term formulae includes term equalities $T \doteq U$ and is closed under the usual propositional connectives.

4.2.1 Lattice Calculus

Equality in a power-set lattice is axiomatized.

Definition 4.1 (*LC*) The (power set) lattice calculus *LC* is the logical system for the language of term formulae consisting of the following axiom schemata:

All substitution instances of tautologies of propositional calculus		**Prop**
$T \doteq T$	equivalence (reflexivity)	T1
$T \doteq U \rightarrow U \doteq T$	equivalence (symmetry)	T2
$T \doteq U \wedge U \doteq V \rightarrow T \doteq V$	equivalence (transitivity)	T3
$T \doteq U \wedge S \doteq V \rightarrow S \sqcup T \doteq V \sqcup U$	join-congruence	T4
$T \doteq U \wedge S \doteq V \rightarrow S \sqcap T \doteq V \sqcap U$	meet-congruence	T5
$T \sqcup U \doteq U \sqcup T$	join-commutativity	T6
$T \sqcap U \doteq U \sqcap T$	meet-commutativity	T7
$(T \sqcup U) \sqcup V \doteq T \sqcup (U \sqcup V)$	join-associativity	T8
$(T \sqcap U) \sqcap V \doteq T \sqcap (U \sqcap V)$	meet-associativity	T9
$T \sqcup (T \sqcap U) \doteq T$	meet-absorption	T10
$T \sqcap (T \sqcup U) \doteq T$	join-absorption	T11
$T \sqcap (U \sqcup V) \doteq (T \sqcap U) \sqcup (T \sqcap V)$	distributivity	T12
$\{\alpha_1, \ldots, \alpha_n\} \doteq \{\alpha_1\} \sqcup \cdots \sqcup \{\alpha_n\}$	atomicity	T13
$\{\alpha\} \doteq \{\beta\} \rightarrow \{\alpha\} \sqcap \{\beta\} \doteq \{\alpha\}$		T14
$\neg(\{\alpha\} \doteq \{\beta\}) \rightarrow \{\alpha\} \sqcap \{\beta\} \doteq \emptyset$		T15

and the following transformation rule

$$\frac{\vdash \phi, \vdash \phi \rightarrow \psi}{\vdash \psi} \qquad \textbf{MP}$$

□

Note that in addition to the axiomatization of the distributive lattice properties (T1–T12), we also need to "connect" non-basic terms to the basic terms they (intuitively) denote. For example, we must have that $\{\alpha\} \sqcup \{\beta\} \doteq \{\alpha, \beta\}$. The axioms T13–T15 ensure this.

Some properties of *LC* are now shown. First, some well-known properties of distributive lattices (the first twelve hold for general lattices).

Lemma 4.1

1. $\vdash T \sqcup T \doteq T$
2. $\vdash T \sqcap T \doteq T$
3. $\vdash T \preceq T \sqcup U$
4. $\vdash T \sqcap U \preceq T$
5. $\vdash U \preceq T \sqcup U$
6. $\vdash T \sqcap U \preceq U$
7. $\vdash T \preceq V \land U \preceq V \to T \sqcup U \preceq V$
8. $\vdash V \preceq T \land V \preceq U \to V \preceq T \sqcap U$
9. $\vdash T \preceq V \land U \preceq W \to T \sqcup U \preceq V \sqcup W$
10. $\vdash T \preceq V \land U \preceq W \to T \sqcap U \preceq V \sqcap W$
11. $\vdash U \preceq T \leftrightarrow T \sqcap U \doteq U$
12. $\vdash T \doteq U \leftrightarrow T \preceq U \land U \preceq T$
13. $\vdash T \sqcup (U \sqcap V) \doteq (T \sqcup U) \sqcap (T \sqcup V)$ □

PROOF See Appendix B. ■

The following Lemma justifies the earlier comments regarding the set notation (recall the underlined notation from Sec. 3.4).

Lemma 4.2 If the sequence $\beta_1 \ldots \beta_m$ is a permutation, possibly with duplicates, of a sequence $\alpha_1 \ldots \alpha_k$ of agent-language formulae, then

$$\vdash \underline{\{}\alpha_1\underline{,} \ldots \underline{,}\alpha_k\underline{\}} \doteq \underline{\{}\beta_1\underline{,} \ldots \underline{,}\beta_m\underline{\}}$$ □

PROOF Follows by T1–T4, T6, T8, T13, and Lemma 4.1.1. ■

Thus, in the notation of a basic set, the order in which we write down the formulae does not matter (with respect to equality).

4.2.2 Equality and Inequality of Basic Terms

In order to complete the term calculus, an axiomatization of equality and inequality of basic terms, i.e. of agent language formulae, is constructed. In short, equal formulae differ at most by the the occurrence of (syntactically) different terms which can be proven equal.

Definition 4.2 (*TC*) The term calculus *TC* is the logical system for the language of term formulae consisting of the lattice calculus *LC* in addition to the following axiom schemata:

$$\begin{array}{lll}
\{\alpha_1,\ldots,\alpha_k\} \preceq \{\alpha'_1,\ldots,\alpha'_m\} \to & ((\{\alpha_1\} \doteq \{\alpha'_1\} \vee \cdots \vee \{\alpha_1\} \doteq \{\alpha'_m\})\wedge & \text{N1} \\
 & (\{\alpha_2\} \doteq \{\alpha'_1\} \vee \cdots \vee \{\alpha_2\} \doteq \{\alpha'_m\})\wedge & \\
 & \vdots & \\
 & (\{\alpha_k\} \doteq \{\alpha'_1\} \vee \cdots \vee \{\alpha_k\} \doteq \{\alpha'_m\})) & \\
\neg(\{p\} \doteq \{\alpha\}) & \text{if } \alpha \neq p & \text{N2} \\
T \doteq S \leftrightarrow \{\triangle_i T\} \doteq \{\triangle_i S\} & & \text{N3} \\
\neg(\{\triangle_i T\} \doteq \{\alpha\}) & \text{if } \alpha \neq \triangle_i S & \text{N4} \\
T \doteq S \leftrightarrow \{\bigtriangledown_i T\} \doteq \{\bigtriangledown_i S\} & & \text{N5} \\
\neg(\{\bigtriangledown_i T\} \doteq \{\alpha\}) & \text{if } \alpha \neq \bigtriangledown_i S & \text{N6} \\
\{\alpha\} \doteq \{\beta\} \leftrightarrow \{\neg\alpha\} \doteq \{\neg\beta\} & & \text{N7} \\
\neg(\{\neg\alpha\} \doteq \{\beta\}) & \text{if } \beta \neq \neg\gamma & \text{N8} \\
\{\alpha_1\} \doteq \{\beta_1\} \wedge \{\alpha_2\} \doteq \{\beta_2\} \leftrightarrow & \{(\alpha_1 \wedge \alpha_2)\} \doteq \{(\beta_1 \wedge \beta_2)\} & \text{N9} \\
\neg(\{(\alpha_1 \wedge \alpha_2)\} \doteq \{\beta\}) & \text{if } \beta \neq (\beta_1 \wedge \beta_2) & \text{N10}
\end{array}$$

□

N2–N10 axiomatize equality and inequality for all possible combinations of singular basic terms. N1 characterizes inequality of non-singular basic terms, in terms of inequality of singular basic terms (note that *equality* of non-singular basic terms is defined by the equivalence and congruence axioms together with the axiomatization of equivalence of singular basic terms, so the other direction of the implication N1 also holds).

4.2.3 Soundness and Completeness of the Term Calculus

Recall the definition of validity of term formulae (Def. 3.4).

Lemma 4.3 (Soundness of *TC*)

$$\vdash_{TC} \phi \Rightarrow \models \phi$$

□

PROOF All axioms of the schema **Prop** are valid, since the interpretation of the propositional connectives is the same as in propositional logic. Given the interpretation of AL (Def. 3.7) and the interpretation of $\doteq$ as equality and $\sqcup$ and $\sqcap$ as set union and intersection, it is obvious that each of the axioms T1–T15 and N1–N10 are valid. Clearly, if $\models \phi_1$ and $\models \phi_1 \to \phi_2$ then $\models \phi_2$, so **MP** preserves validity. Let $\vdash \phi$. $\models \phi$ follows by a simple inductive proof over the length of the proof of $\vdash \phi$. ■

The following are some definitions and results which are used to prove completeness of the term calculus. The exposition is quite technical, and the reader may want to jump directly to the main result in Corollary 4.2 (the immediately following discussion about term terminology may, however, be useful to review).

The following terminology is useful for inductive proofs over the structure of formulae. The *degree* $d(T)$ of a term $T \in TL$ is 1 plus the highest number of nested knowledge operators inside the term, and is defined as follows:

- $d(T \sqcup U) = \max(d(T), d(U))$
- $d(T \sqcap U) = \max(d(T), d(U))$
- $d(\{\alpha_1, \dots, \alpha_k\}) = \max(d(\{\alpha_1\}), \dots, d(\{\alpha_k\}))$
- $d(\{p\}) = 1$
- $d(\{\triangle_i S\}) = 1 + d(S)$
- $d(\{\triangledown_i S\}) = 1 + d(S)$
- $d(\{\neg\alpha\}) = d(\{\alpha\})$
- $d(\{(\alpha \land \beta)\}) = \max(d(\{\alpha\}), d(\{\beta\}))$

S is a subterm of a term T iff it is a term and it occurs as a substring in T, and it is a proper subterm iff it in addition is different from T. A subterm S occurs at a certain *level* $l_T(S)$[1] in the term T, corresponding to the number of nested knowledge operators up to the point of the occurrence. The level of S in T is defined as follows:

$$l_T(S) = d(T) - d(S) + 1$$

As an illustration, the following term T of degree 3 is annotated by the level in T (above) and degree (below) of each subterm:

$$T = \overbrace{\underbrace{\{\triangle_i \overbrace{\underbrace{\{p, \triangledown_i \overbrace{\underbrace{\overbrace{\underbrace{\{q\}}_{1}}^{3} \sqcap \overbrace{\underbrace{\{q,r\}}_{1}}^{3}}_{1}}^{3}\}}_{2}}^{2}, \triangledown_i \overbrace{\underbrace{\{\triangle_j \overbrace{\underbrace{\{p\}}_{1}}^{3}\}}_{2}}^{2}\}}_{3}}^{1}$$

In the following lemmata, the notation $S[\cdot]$ and $\phi[\cdot]$ for a term and a formula respectively with a *hole*, i.e. with a *missing term*, is used. If T is a term, then $S[T]$ ($\phi[T]$) is a well-formed term (formula)[2].

A hole can occur at any place in a term or formula where a well-formed term can occur, hence a hole occurs at a certain level in the term or formula – and it may occur at *any* level. Henceforth, a hole "at level n" means a missing occurrence of a subterm at level n.

I now show that we can replace a subterm with an *equal* subterm, at an arbitrary level inside a term.

[1]Of course, the same subterm may occur at different levels in the term. Formally, we can view each occurrence of a subterm as being annotated by a unique name. In other words, "the level of S in T" is the level of a particular occurrence of S in T.

[2]Note that the notation for a missing term and for the interpretation of a term both use the [] operator, but can not be confused. For example, $[S[T]]$ means the interpretation of the term resulting from replacing the "hole" in $S[\cdot]$ with T.

Lemma 4.4 (Substitution) The following transformation rule is admissible in *TC*:

$$\frac{\vdash T \doteq U}{\vdash S[T] \doteq S[U]} \qquad \textbf{Repl}$$

□

PROOF I first show an intermediate result; that we can substitute equal terms on the first level, i.e. that the following rule is admissible:

$$\text{If } S[\cdot] \text{ has a hole at level 1 then } \frac{\vdash T \doteq U}{\vdash S[T] \doteq S[U]} \qquad \textbf{Repl}^-$$

Terms with holes at level 1 can be defined in the following way. $[\cdot]$ is a term with a hole at level 1, and if $S[\cdot]$ is a term with a hole at level 1 and V is a term then $S[\cdot] \sqcup V$, $V \sqcup S[\cdot]$, $S[\cdot] \sqcap V$, $V \sqcap S[\cdot]$ are terms with holes at level 1. The proof is by structural induction over $S[\cdot]$. Assume $\vdash T \doteq U$. In the base case, $S[\cdot]$ is empty and $S[T] = T$, and the consequent of the transformation rule is the same as the antecedent. For the induction step, $S[\cdot]$ is constructed by combining a term with a hole with a well-formed term using $\sqcup$ or $\sqcap$. First, consider the case where $S[\cdot] = S_1[\cdot] \sqcup S_2$, i.e. where the hole in S is in the left sub term:

1	$\vdash T \doteq U$	Assumption
2	$\vdash S_1[T] \doteq S_1[U]$	Ind. hyp.(1)
3	$\vdash S_2 \doteq S_2 \land S_1[T] \doteq S_1[U] \to S_1[T] \sqcup S_2 \doteq S_1[U] \sqcup S_2$	T4
4	$\vdash S_2 \doteq S_2$	T1
5	$\vdash S_2 \doteq S_2 \land S_1[T] \doteq S_1[U]$	**MP**(**Prop**,2,4)
6	$\vdash S_1[T] \sqcup S_2 \doteq S_1[U] \sqcup S_2$	**MP**(5,3)

Next, consider $S[\cdot] = S_1[\cdot] \sqcap S_2$:

1	$\vdash T \doteq U$	Assumption
2	$\vdash S_1[T] \doteq S_1[U]$	Ind. hyp.(1)
3	$\vdash S_2 \doteq S_2 \land S_1[T] \doteq S_1[U] \to S_1[T] \sqcap S_2 \doteq S_1[U] \sqcap S_2$	T5
4	$\vdash S_2 \doteq S_2$	T1
5	$\vdash S_2 \doteq S_2 \land S_1[T] \doteq S_1[U]$	**MP**(**Prop**,2,4)
6	$\vdash S_1[T] \sqcap S_2 \doteq S_1[U] \sqcap S_2$	**MP**(5,3)

The proofs in the cases where $S[\cdot] = S_1 \sqcup S_2[\cdot]$ and $S[\cdot] = S_1 \sqcap S_2[\cdot]$ are similar. This completes the proof of **Repl**$^-$.

The proof of **Repl** is by induction over the level of the hole in $S[\cdot]$. Let $\vdash T \doteq U$. For the induction base, consider a term $S[\cdot]$ with a hole in level 1. Then $\vdash S[T] \doteq S[U]$ by **Repl**$^-$.

For the inductive step, assume that **Repl** holds for all terms with holes at level k. Let $S[\cdot]$ be a term with a hole at level $k+1$. Then $S[\cdot]$ contains a level k term on the form $V[\cdot] = \{\alpha_1, \ldots, \alpha_j[\cdot], \ldots, \alpha_m\}$, where $\alpha_j[\cdot]$ contains a term with a hole on level 1. I now show that

$$\vdash \{\alpha_j[T]\} \doteq \{\alpha_j[U]\} \tag{4.1}$$

by induction over the structure of $\alpha_j[\cdot]$. For the base case, first let $\alpha_j[\cdot] = \triangle_i W[\cdot]$ where $W[\cdot]$ is a term with a missing level 1 subterm. By **Repl**$^-$, $\vdash W[T] \doteq W[U]$, and thus $\vdash \{\triangle_i W[T]\} \doteq \{\triangle_i W[U]\}$ by N3. Second, let $\alpha_j[\cdot] = \triangledown_i W[\cdot]$, and $\vdash \{\triangledown_i W[T]\} \doteq \{\triangledown_i W[U]\}$ follows similarly by **Repl**$^-$ and N5. For the induction step, first let $\alpha_j[\cdot] = \neg\beta[\cdot]$. By the induction hypothesis, $\vdash \{\beta[T]\} \doteq \{\beta[U]\}$ and thus $\vdash \{\neg\beta[T]\} \doteq \{\neg\beta[U]\}$ by N7. Second, let $\alpha_j[\cdot] = (\beta_1[\cdot] \wedge \beta_2)$. By the induction hypothesis $\vdash \{\beta_1[T]\} \doteq \{\beta_1[U]\}$, and by T1 $\vdash \{\beta_2\} \doteq \{\beta_2\}$, and thus $\vdash \{(\beta_1[T] \wedge \beta_2)\} \doteq \{(\beta_1[U] \wedge \beta_2)\}$ by N9. Third, let $\alpha_j[\cdot] = (\beta_1 \wedge \beta_2[\cdot])$ and $\vdash \{(\beta_1 \wedge \beta_2[T])\} \doteq \{(\beta_1 \wedge \beta_2[U])\}$ follows by similar reasoning. This completes the proof of (4.1). Then, it follows from several applications of T13 and T4 that $\vdash V[T] \doteq V[U]$. If we replace the term $V[\cdot]$ in $S[\cdot]$ with a hole, we get a term $S'[\cdot]$ with a missing k level term. From $\vdash V[T] \doteq V[U]$ and the induction hypothesis, we have that $\vdash S'[V[T]] \doteq S'[V[U]]$. Clearly, $S'[V[T]] = S[T]$ and $S'[V[U]] = S[U]$ and thus we have that $\vdash S[T] \doteq S[U]$ which completes the inductive step in the main proof. ■

Completeness of the term calculus follows as a corollary of the next lemma, the proof of which is quite involved and found in an appendix.

Lemma 4.5 Let T_1 and T_2 be terms.

1. If $[T_1] = [T_2]$ then $\vdash T_1 \doteq T_2$
2. If $[T_1] \neq [T_2]$ then $\vdash \neg T_1 \doteq T_2$ □

PROOF See Appendix C. ■

Corollary 4.1 Let T and U be terms.

1. If $[T] \subseteq [U]$ then $\vdash T \preceq U$
2. If $[T] \not\subseteq [U]$ then $\vdash \neg T \preceq U$ □

PROOF If $[T] \subseteq [U]$ then $[T \sqcup U] = [T] \cup [U] = [U]$, and $\vdash T \sqcup U \doteq U$ by Lemma 4.5.1. If $[T] \not\subseteq [U]$ then $[T \sqcup U] = [T] \cup [U] \neq [U]$, and $\vdash \neg T \sqcup U \doteq U$ by Lemma 4.5.2. ■

Finally, completeness of the term calculus is shown.

Corollary 4.2 If ϕ is a term formula, then

$$\models \phi \Rightarrow \vdash \phi$$ □

PROOF The following statement is proved by structural induction over ϕ:

$$\text{If } \models \phi \text{ then } \vdash \phi, \text{ otherwise } \vdash \neg\phi \tag{4.2}$$

Recall the definition of validity of term formulae (Def. 3.4). In the base case, $\phi = T \doteq U$. If $\models \phi$ then $[T] = [U]$ and $\vdash T \doteq U$ by Lemma 4.5.1. If $\not\models \phi$ then $[T] \neq [U]$ and $\vdash \neg T \doteq U$ by Lemma 4.5.2. For the induction step, first assume that $\phi = \neg\psi$. If $\models \neg\psi$ then $\not\models \psi$ and $\vdash \neg\psi$ by the induction hypothesis. If $\not\models \neg\psi$ then $\models \psi$, $\vdash \psi$ by the induction hypothesis and $\vdash \neg\neg\psi$ by **Prop**. Second, assume that $\phi = (\psi_1 \wedge \psi_2)$. If $\models \phi$, $\models \psi_1$ and $\models \psi_2$, $\vdash \psi_1$ and $\vdash \psi_2$ by the induction hypothesis and $\vdash (\psi_1 \wedge \psi_1)$ by **Prop**. If $\not\models \phi$ then $\not\models \psi_1$ or $\not\models \psi_2$, $\vdash \neg\psi_1$ or $\vdash \neg\psi_2$ by the induction hypothesis, and thus $\vdash \neg(\psi_1 \wedge \psi_2)$ by **Prop**. ■

4.3 Epistemic Calculus

Finally, a logical system for the epistemic language is introduced. The epistemic language includes the language of term formulae, and the epistemic calculus includes the term calculus.

Definition 4.3 (*EC*) The epistemic calculus *EC* is the logical system for the epistemic language *EL* consisting of the term calculus *TC* in addition to the following axiom schemata:

All substitution instances of tautologies of propositional calculus	**Prop**
$\triangle_i \emptyset$	E1
$(\triangle_i T \wedge \triangle_i U) \rightarrow \triangle_i (T \sqcup U)$	E2
$(\triangledown_i T \wedge \triangledown_i U) \rightarrow \triangledown_i (T \sqcap U)$	E3
$(\triangle_i T \wedge \triangledown_i U) \rightarrow T \preceq U$	E4
$(\triangledown_i (U \sqcup \{\alpha\}) \wedge \neg \triangle_i \{\alpha\}) \rightarrow \triangledown_i U$	E5
$\triangle_i T \wedge U \preceq T \rightarrow \triangle_i U$	**KS**
$\triangledown_i T \wedge T \preceq U \rightarrow \triangledown_i U$	**KG**

and the following transformation rule

$$\frac{\Gamma \vdash \phi, \Gamma \vdash \phi \rightarrow \psi}{\Gamma \vdash \psi} \qquad \textbf{MP}$$

□

Example 4.1 Consider again the encrypted message example in the introduction (p. 27). In Example 3.1 (p. 35) the following formulae were used to represent sentences (3.2), (3.3), (3.5) and (3.4):

$$\begin{aligned} \triangle_b \{m_e\} \\ \triangle_b \{n\} \\ \triangledown_b \{m_e, n\} \\ \neg \triangle_b \{m\} \end{aligned}$$

The following in an example proof in the calculus, of:

$$\{\triangle_b \{m_e\}, \triangle_b \{n\}, \triangledown_b \{m_e, n\}\} \vdash \neg \triangle_b \{m\}$$

Let $\Gamma = \{\triangle_b\{m_e\}, \triangle_b\{n\}, \triangledown_b\{m_e, n\}\}$.

1	$\Gamma \vdash ((\triangle_b\{m\} \wedge \triangledown_b\{m_e,n\}) \to \{m\} \preceq \{m_e,n\})$ $\to ((\triangledown_b\{m_e,n\} \wedge \neg\{m\} \preceq \{m_e,n\}) \to \neg \triangle_b \{m\})$	**Prop**
2	$\Gamma \vdash (\triangle_b\{m\} \wedge \triangledown_b\{m_e,n\}) \to \{m\} \preceq \{m_e,n\}$	E4
3	$\Gamma \vdash (\triangledown_b\{m_e,n\} \wedge \neg\{m\} \preceq \{m_e,n\}) \to \neg \triangle_b \{m\}$	**MP**(2,3)
4	$\Gamma \vdash \neg(\{m\} \preceq \{m_e,n\})$	Cor. 4.1.2
5	$\Gamma \vdash \triangledown_b\{m_e,n\}$ $\to (\neg\{m\} \preceq \{m_e,n\} \to (\triangledown_b\{m_e,n\} \wedge \neg\{m\} \preceq \{m_e,n\}))$	**Prop**
6	$\Gamma \vdash \triangledown_b\{m_e,n\}$	Premise
7	$\Gamma \vdash \neg\{m\} \preceq \{m_e,n\} \to (\triangledown_b\{m_e,n\} \wedge \neg\{m\} \preceq \{m_e,n\})$	**MP**(5,6)
8	$\Gamma \vdash \triangledown_b\{m_e,n\} \wedge \neg\{m\} \preceq \{m_e,n\}$	**MP**(4,7)
9	$\Gamma \vdash \neg \triangle_b \{m\}$	**MP**(3,8) $\square$

In Example 4.1 every deduction in the proof were explicitly written down, except in line (4). I will henceforth often drop trivial details in the proofs, by e.g. combining several applications of **Prop** and **MP** in one line. Since the set of term formulae (from the previous section) is a subset of *EL*, I will, as in line (4) above, refer to the completeness of the term calculus instead of writing out the proofs of term (in)equalities.

The deduction theorem, $\Gamma \cup \{\gamma\} \vdash \phi \Rightarrow \Gamma \vdash \gamma \to \phi$, clearly holds for *EC*, and will be referred to as *DT*.

4.3.1 Soundness and Completeness

I show that *EC* is sound and complete[3]. Note that since *EC* extends *TC*, if $\Gamma \vdash \phi$ and Γ is consistent and ϕ is a term formula then $\models \phi$ by completeness of *TC*: if $\not\models \phi$ then $\models \neg\phi$ and $\vdash \neg\phi$ by completeness and thus Γ is not consistent.

Theorem 4.1 (Soundness)

$$\text{If } \Gamma \vdash \phi \text{ then } \Gamma \models \phi \qquad \square$$

PROOF I show that all axioms of *EC* are valid (we only need to consider the "new" axioms in Definition 4.3; all the axioms of *TC* are valid since *TC* is sound). Let $M = (s_1, \ldots, s_n, \pi)$ be an arbitrary general knowledge set structure. I show that $M \models \phi$ for each axiom ϕ.

Prop All **Prop**-axioms are valid, since the interpretation of the propositional connectives is the same as in propositional logic.

E1 $\phi = \triangle_i\emptyset$. $[\emptyset] = \emptyset \subseteq s_i \Rightarrow M \models \phi$.

E2 $\phi = (\triangle_i T \wedge \triangle_i U) \to \triangle_i(T \sqcup U)$. If $M \not\models \triangle_i T \wedge \triangle_i U$, then $M \models \phi$ trivially. If $M \models \triangle_i T \wedge \triangle_i U$, then

$$\left.\begin{array}{l} M \models \triangle_i T \Rightarrow [T] \subseteq s_i \\ M \models \triangle_i U \Rightarrow [U] \subseteq s_i \end{array}\right\} \Rightarrow [T \sqcup U] = [T] \cup [U] \subseteq s_i \Rightarrow M \models \triangle_i(T \sqcup U)$$

and thus $M \models \phi$.

[3]Soundness and completeness of *EC* relies on soundness and completeness of *TC*. As a generalization, *any* sound and complete calculus of term (in)equality could be used instead of *TC*.

E3 $\phi = (\triangledown_i T \wedge \triangledown_i U) \to \triangledown_i (T \sqcap U)$. If $M \not\models \triangledown_i T \wedge \triangledown_i U$, then $M \models \phi$ trivially. If $M \models \triangledown_i T \wedge \triangledown_i U$, then

$$\left.\begin{array}{l} M \models \triangledown_i T \Rightarrow s_i \subseteq [T] \\ M \models \triangledown_i U \Rightarrow s_i \subseteq [U] \end{array}\right\} \Rightarrow s_i \subseteq [T] \cap [U] = [T \sqcap U] \Rightarrow M \models \triangledown_i (T \sqcap U)$$

and thus $M \models \phi$.

E4 $\phi = (\triangle_i T \wedge \triangledown_i U) \to T \preceq U$. If $M \not\models \triangle_i T \wedge \triangledown_i U$, then $M \models \phi$ trivially. If $M \models \triangle_i T \wedge \triangledown_i U$, then

$$\left.\begin{array}{l} M \models \triangle_i T \Rightarrow [T] \subseteq s_i \\ M \models \triangledown_i U \Rightarrow s_i \subseteq [U] \end{array}\right\} \Rightarrow [T] \subseteq [U] \Rightarrow M \models T \preceq U$$

and thus $M \models \phi$.

E5 $\phi = (\triangledown_i (U \sqcup \{\alpha\}) \wedge \neg \triangle_i \{\alpha\}) \to \triangledown_i U$. If $M \not\models \triangledown_i (U \sqcup \{\alpha\}) \wedge \neg \triangle_i \{\alpha\}$, then $M \models \phi$ trivially. If $M \models \triangledown_i (U \sqcup \{\alpha\}) \wedge \neg \triangle_i \{\alpha\}$, then

$$\left.\begin{array}{l} M \models \triangledown_i (U \sqcup \{\alpha\}) \Rightarrow s_i \subseteq [U \sqcup \{\alpha\}] = [U] \cup \{[\alpha]\} \\ M \models \neg \triangle_i \{\alpha\} \Rightarrow [\alpha] \notin s_i \end{array}\right\} \Rightarrow s_i \subseteq [U] \Rightarrow M \models \triangledown_i U$$

and thus $M \models \phi$.

KS $\phi = (\triangle_i T \wedge U \preceq T) \to \triangle_i U$. If $M \not\models \triangle_i T \wedge U \preceq T$, then $M \models \phi$ trivially. If $M \models \triangle_i T \wedge U \preceq T$, then

$$\left.\begin{array}{l} M \models \triangle_i T \Rightarrow [T] \subseteq s_i \\ M \models U \preceq T \Rightarrow [U] \subseteq [T] \end{array}\right\} \Rightarrow [U] \subseteq s_i \Rightarrow M \models \triangle_i U$$

and thus $M \models \phi$.

KG $\phi = (\triangledown_i T \wedge T \preceq U) \to \triangledown_i U$. If $M \not\models \triangledown_i T \wedge T \preceq U$, then $M \models \phi$ trivially. If $M \models \triangledown_i T \wedge T \preceq U$, then

$$\left.\begin{array}{l} M \models \triangledown_i T \Rightarrow s_i \subseteq [T] \\ M \models T \preceq U \Rightarrow [T] \subseteq [U] \end{array}\right\} \Rightarrow s_i \subseteq [U] \Rightarrow M \models \triangledown_i U$$

and thus $M \models \phi$.

Clearly, if $\Gamma \models \phi_1$ and $\Gamma \models \phi_1 \to \phi_2$ then $\Gamma \models \phi_2$, so **MP** preserves logical consequence.

Let $\Gamma \vdash \phi$. $\Gamma \models \phi$ follows by a simple inductive proof over the length of the proof of $\Gamma \vdash \phi$. ■

In order to show completeness, I first show the counterpart of Lemma 4.4 for *EC* — we can substitute equals for equals inside an *EL*-formula.

Lemma 4.6 (Substitution) The following transformation rule is admissible in *EC*:

$$\frac{\Gamma \vdash T \doteq U}{\Gamma \vdash \phi[T] \leftrightarrow \phi[U]} \quad \textbf{Subst}$$

□

PROOF The Lemma holds trivially for inconsistent Γ, so assume that Γ is consistent. The proof is by structural induction over $\phi[\cdot]$[4]. The base cases are the three types of atomic formulae involving a term. In the first base case, $\phi[\cdot] = \triangle_i S[\cdot]$:

1	$\Gamma \vdash T \doteq U$	Assumption
2	$\Gamma \vdash S[U] \preceq S[T]$	Compl. of *TC* (1)
3	$\Gamma \vdash \triangle_i S[T] \wedge S[U] \preceq S[T] \rightarrow \triangle_i S[U]$	**KS**
4	$\Gamma \vdash (\triangle_i S[T] \wedge S[U] \preceq S[T] \rightarrow \triangle_i S[U])$ $\rightarrow (S[U] \preceq S[T] \rightarrow (\triangle_i S[T] \rightarrow \triangle_i S[U]))$	**Prop**
5	$\Gamma \vdash \triangle_i S[T] \rightarrow \triangle_i S[U]$	**MP**(2,3,4)

The proof for $\Gamma \vdash \triangle_i S[U] \rightarrow \triangle_i S[T]$ can be obtained by using T2 with line 1, and exchanging $S[T]$ and $S[U]$ in the rest of the proof above. In the second base case, $\phi[\cdot] = \triangledown_i S[\cdot]$, and the proof is obtained by replacing the three last lines of the preceding proof with:

3	$\Gamma \vdash \triangledown_i S[U] \wedge S[U] \preceq S[T] \rightarrow \triangledown_i S[T]$	**KG**
4	$\Gamma \vdash (\triangledown_i S[U] \wedge S[U] \preceq S[T] \rightarrow \triangledown_i S[T])$ $\rightarrow (S[U] \preceq S[T] \rightarrow (\triangledown_i S[U] \rightarrow \triangledown_i S[T]))$	**Prop**
5	$\Gamma \vdash \triangledown_i S[U] \rightarrow \triangledown_i S[T]$	**MP**(2,3,4)

and similarly for $\Gamma \vdash \triangledown_i S[T] \rightarrow \triangledown_i S[U]$.

Consider now the third base case where $\phi[\cdot] = (S[\cdot] \doteq V)$ (the proof in the case where $\phi[\cdot] = (S \doteq V[\cdot])$ is similar):

1	$\Gamma \vdash T \doteq U$	Assumption
2	$\Gamma \vdash S[U] \doteq S[T]$	Compl. of *TC* (1)
3	$\Gamma \vdash S[U] \doteq S[T] \wedge S[T] \doteq V \rightarrow S[U] \doteq V$	T3
4	$\Gamma \vdash S[U] \doteq S[T] \rightarrow (S[T] \doteq V \rightarrow S[U] \doteq V)$	**MP**,**Prop**, 3
5	$\Gamma \vdash S[T] \doteq V \rightarrow S[U] \doteq V$	**MP**,2,4

The proof of $\Gamma \vdash S[U] \doteq V \rightarrow S[T] \doteq V$ can be obtained by using T2 with line (1), and exchanging $S[T]$ and $S[U]$ in the rest of the proof above.

In the inductive step, first consider the case where $\phi[\cdot] = \neg\psi[\cdot]$. By the induction hypothesis, $\Gamma \vdash \psi[T] \leftrightarrow \psi[U]$ whenever $\Gamma \vdash T \doteq U$, and $\Gamma \vdash \neg\psi[T] \leftrightarrow \neg\psi[U]$ follows by **Prop**. Second, consider the case where $\phi[\cdot] = \psi_1[\cdot] \wedge \psi_2$ (the proof in the case that $\phi[\cdot] = \psi_1 \wedge \psi_2[\cdot]$ is similar). By the induction hypothesis, $\Gamma \vdash \psi_1[T] \leftrightarrow \psi_1[U]$ whenever $\Gamma \vdash T \doteq U$, and $\Gamma \vdash \psi_1[T] \wedge \psi_2 \leftrightarrow \psi_1[U] \wedge \psi_2$ by **Prop**. Clearly, $\psi_1[T] \wedge \psi_2 = (\psi_1 \wedge \psi_2)[T]$ (the hole in $(\psi_1 \wedge \psi_2)[\cdot]$ is the hole in ψ_1), so $\Gamma \vdash (\psi_1 \wedge \psi_2)[T] \leftrightarrow (\psi_1 \wedge \psi_2)[U]$. ■

Completeness of *EL*, with respect to $\mathcal{M}$, is now shown by the commonly used strategy of showing satisfiability of maximal consistent theories. It may be instructive to recall that for every finite set $X \in \wp^{fin}(OL)$, there exists a term T such that $[T] = X$.

Lemma 4.7 If Γ is a consistent theory, then there exists a maximal consistent theory $\Gamma' \supseteq \Gamma$. □

[4]Formulae and terms with holes were discussed in Section 4.2.3.

PROOF $\Gamma' \supseteq \Gamma$ is constructed as follows. EL is countable, so let $\phi_1, \phi_2, \ldots$ be an enumeration of positive formulae from EL (i.e., those where the outermost operator is not $\neg$):

$$\Gamma_0 = \Gamma \tag{4.3}$$

$$\Gamma_{i+1} = \begin{cases} \Gamma_i \cup \{\phi_{i+1}\} & \text{if } \Gamma_i \cup \{\phi_{i+1}\} \text{ is consistent} \\ \Gamma_i \cup \{\neg\phi_{i+1}\} & \text{otherwise} \end{cases} \tag{4.4}$$

$$\Gamma' = \bigcup_{i=0}^{\infty} \Gamma_i \tag{4.5}$$

Since Γ_0 is consistent, clearly all Γ_i are consistent. Let Δ be a finite subset of Γ'. $\Delta \subseteq \Gamma_j$ for some j, so Δ must be consistent. Since all finite subsets of Γ' are consistent, so is Γ'. Since either $\Gamma' \models \phi$ or $\Gamma' \models \neg\phi$ for any ϕ, Γ' is maximal. ■

Lemma 4.8 Let Γ' be a maximal consistent theory. If there exists a T' such that $\Gamma' \vdash \triangledown_i T'$, then there exists a T such that $\Gamma' \vdash \Diamond_i T$. □

PROOF Let $\Gamma' \vdash \triangledown_i T'$. Let T be a term such that

$$[T] = \bigcap_{[S] \subseteq [T'] \text{ and } \Gamma' \vdash \triangledown_i S} [S]$$

(this set is finite since it is included in $[T']$, and thus such a term T exists). Since $[T']$ is finite it has only a finite number of subsets, and thus $\Gamma' \vdash \triangledown_i T$ can be obtained by a finite number of applications of E3, completeness of the term calculus and **Subst**. Let U be a term such that

$$[U] = \bigcup_{\Gamma' \vdash \triangle_i S} [S]$$

To see that this set indeed is finite, and thus that the term U exists, observe that for any term S such that $\Gamma' \vdash \triangle_i S$, $[S] \subseteq [T]$ by E4, completeness of the term calculus and consistency of Γ', so there can only be a finite number of such terms. By a finite number of applications of E2, completeness of the term calculus and **Subst**, $\Gamma' \vdash \triangle_i U$. By E4 we then have that $\Gamma' \vdash U \preceq T$ and thus $[U] \subseteq [T]$ by consistency of Γ' and completeness of the term calculus.

I now show that $[T] \subseteq [U]$. Assume the opposite; that there is a x such that $x \in [T]$ and $x \notin [U]$. Then there is a α such that $[\alpha] = x$. Let T^- be a term such that

$$[T^-] = [T] \setminus [\{\alpha\}]$$

(T^- must exist since $[T]$ is finite). $[T] = [T^- \sqcup \{\alpha\}]$ and by completeness of the term calculus $\Gamma' \vdash T \doteq T^- \sqcup \{\alpha\}$. Since $\Gamma' \vdash \triangledown_i T$, $\Gamma' \vdash \triangledown_i (T^- \sqcup \{\alpha\})$ by **Subst**. If $\Gamma' \vdash \triangle_i \{\alpha\}$ then $[\alpha] \in U$ which is a contradiction, hence $\Gamma' \vdash \neg \triangle_i \{\alpha\}$ by the maximality of Γ'. By E5, $\Gamma' \vdash \triangledown_i T^-$. But then $[T] \subseteq [T^-]$ by the construction of T, which is a contradiction. Therefore, $[T] \subseteq [U]$. Thus $[T] = [U]$ and, by completeness of the term calculus, $\Gamma' \vdash T \doteq U$ and, by **Subst**, $\Gamma' \vdash \Diamond_i T$. ■

Lemma 4.9 Every maximal consistent theory is satisfiable. □

PROOF Let Γ be a maximal consistent theory. The following general knowledge set structure is constructed:

$$M^\Gamma = (s_1^\Gamma, \ldots, s_2^\Gamma, \pi^\Gamma)$$

where

$$\pi^\Gamma(p) = \textbf{true} \Leftrightarrow \Gamma \vdash p$$

$$s_i^\Gamma = \begin{cases} [W] \text{ where } \Gamma \vdash \Diamond_i W & \text{if there is a } T' \text{ such that } \Gamma \vdash \bigtriangledown_i T' \\ \bigcup_{\Gamma \vdash \triangle_i S}[S] \cup \{*\} & \text{if } \forall_{T'} \Gamma \nvdash \bigtriangledown_i T' \text{ and } \bigcup_{\Gamma \vdash \triangle_i S}[S] \text{ is finite} \\ \bigcup_{\Gamma \vdash \triangle_i S}[S] & \text{if } \forall_{T'} \Gamma \nvdash \bigtriangledown_i T' \text{ and } \bigcup_{\Gamma \vdash \triangle_i S}[S] \text{ is infinite} \end{cases}$$

intended to satisfy Γ. In the definition of s_i^Γ, any W such that $\Gamma \vdash \Diamond_i W$ can be chosen in the case that there exists a T' such that $\Gamma \vdash \bigtriangledown_i T'$; Lemma 4.8 guarantees the existence of such a term. I show, by structural induction over ϕ, that

$$M^\Gamma \models \phi \Longleftrightarrow \Gamma \vdash \phi \tag{4.6}$$

This is a stronger statement than the lemma; the lemma is given by the direction to the left.

For the base case:

- $\phi = p \in \Theta$: $\Gamma \vdash p$ iff $\pi^\Gamma(p) = \textbf{true}$ iff $M^\Gamma \models p$.
- $\phi = T \doteq U$: (4.6) follows by soundness and completeness of TC.
- $\phi = \triangle_i T$: For each direction of (4.6), consider the cases where

  a) there is a T' such that $\Gamma \vdash \bigtriangledown_i T'$ or

  b) $\Gamma \nvdash \bigtriangledown_i T'$ for every T'

  corresponding to the first and to the second and third cases in the definition of s_i^Γ, respectively.

  $\Rightarrow$) If $M^\Gamma \models \triangle_i T$ then $[T] \subseteq s_i^\Gamma$.

  a) $[T] \subseteq [W]$, and $\Gamma \vdash T \preceq W$ by completeness of TC. Since $\Gamma \vdash \triangle_i W$, $\Gamma \vdash \triangle_i T$ by **KS**.

  b) Let $[T] = \{[\alpha_1], \ldots, [\alpha_k]\}$ ($[T] \in \wp^{fin}(OL)$). By the definition of s_i^Γ there exists, for $1 \leq j \leq k$, a proof $\Gamma \vdash \triangle_i T_{\alpha_j}$ where T_{α_j} is a term such that $[\alpha_j] \in [T_{\alpha_j}]$ ($* \notin [S]$ for any term S). Since $[\{\alpha_j\}] \subseteq [T]$, $\Gamma \vdash \{\alpha_j\} \preceq T_{\alpha_j}$ ($1 \leq j \leq k$) by completeness of the term calculus and then $\Gamma \vdash \triangle_i\{\alpha_j\}$ by **KS**. By (repeated applications of) E2, $\Gamma \vdash \triangle_i(\{\alpha_1\} \sqcup \cdots \sqcup \{\alpha_k\})$. Since $[\{\alpha_1\} \sqcup \cdots \sqcup \{\alpha_k\}] = [T]$, $\Gamma \vdash \triangle_i T$ by completeness of the term calculus and **Subst**.

  $\Leftarrow$) **a)** Since $\Gamma \vdash \bigtriangledown_i W$, if $\Gamma \vdash \triangle_i T$ then $\Gamma \vdash T \preceq W$ by E4. If $[T] \nsubseteq [W]$ then $\Gamma \vdash \neg T \preceq W$ by completeness of TC and Γ would be inconsistent, so $[T] \subseteq [W]$ and thus $M^\Gamma \models \triangle_i T$.

  b) If $\Gamma \vdash \triangle_i T$ then $[T] \subseteq s_i^\Gamma$ and thus $M^\Gamma \models \triangle_i T$.

- $\phi = \bigtriangledown_i T$: For each direction of (4.6), consider again the two cases

a) there is a T' such that $\Gamma \vdash \triangledown_i T'$ and

b) $\Gamma \nvdash \triangledown_i T'$ for every T'

$\Rightarrow$) If $M^\Gamma \models \triangledown_i T$ then $s_i^\Gamma \subseteq [T]$.

a) $[W] \subseteq [T]$, and $\Gamma \vdash W \preceq T$ by completeness of TC. Because $\Gamma \vdash \triangledown_i W$, $\Gamma \vdash \triangledown_i T$ by **KG**.

b) If $* \in s_i^\Gamma$, then $* \in [T]$, which is not the case for any T. This case is impossible. If $* \notin s_i^\Gamma$, then s_i^Γ is infinite. But $s_i^\Gamma \subseteq [T]$ and $[T]$ is finite, so this case is also impossible.

$\Leftarrow$) **a)** Since $\Gamma \vdash \triangle_i W$, if $\Gamma \vdash \triangledown_i T$ then $\Gamma \vdash W \preceq T$ by E4. If $[W] \nsubseteq [T]$ then $\Gamma \vdash \neg W \preceq T$ by completeness of TC and Γ would be inconsistent, so $[W] \subseteq [T]$ and thus $M^\Gamma \models \triangledown_i T$.

b) Since $\Gamma \nvdash \triangledown_i T$ for any T, this case is impossible.

For the induction step:

- $\phi = \neg\psi$: $M^\Gamma \models \neg\psi$ iff $M^\Gamma \not\models \psi$ iff (by the induction hypothesis) $\Gamma \nvdash \psi$ iff (by the fact that Γ is both consistent and maximal) $\Gamma \vdash \neg\psi$.

- $\phi = (\psi_1 \wedge \psi_2)$: $M^\Gamma \models (\psi_1 \wedge \psi_2)$ iff $M^\Gamma \models \psi_1$ and $M^\Gamma \models \psi_2$ iff (by the induction hypothesis) $\Gamma \vdash \psi_1$ and $\Gamma \vdash \psi_2$ iff (by **Prop**) $\Gamma \vdash (\psi_1 \wedge \psi_2)$. ■

Lemma 4.10 Every consistent theory is satisfiable. □

PROOF Follows immediately from Lemmas 4.7 and 4.9. ■

Theorem 4.2 If $\Gamma \models \phi$ then $\Gamma \vdash \phi$. □

PROOF Let $\Gamma \models \phi$. Assume that Γ is consistent (otherwise $\Gamma \vdash \phi$ trivially). If $M \models \Gamma$ then $M \models \phi$ and $M \not\models \neg\phi$, so $\Gamma \cup \{\neg\phi\}$ is unsatisfiable and thus inconsistent by Lemma 4.10. Then, $\Gamma \cup \{\neg\phi\} \vdash \phi$, $\Gamma \vdash \neg\phi \rightarrow \phi$ by DT, and $\Gamma \vdash \phi$ by **Prop**. ■

Theorems 4.1 and 4.2 are collected in the following corollary.

Corollary 4.3 For every $\Gamma \subseteq EL$, $\phi \in EL$:

$$\Gamma \models \phi \Leftrightarrow \Gamma \vdash \phi$$ □

Chapter 5

Finite Agents

The motivation behind describing the concept of explicit knowledge is the idea that it is different from implicit knowledge; it is unrealistic to assume that an agent would deduce all possible consequences — generally infinitely many — of his knowledge. In fact, it is not only unrealistic to assume that an agent could *deduce* an infinite amount of facts, but also that he is able to explicitly know (i.e. to store) an infinite amount of facts at all. The latter would require an infinite amount of memory.

In the current chapter, the semantic assumption that agents know only a finite amount of facts at a given time is made. Such agents are henceforth called *finite agents*. In the rest of the thesis, agents are assumed to be finite. The resulting logic is called *Static Syntactic Epistemic Logic (*SSEL*)*.

The restriction to finite epistemic states ensures a concept of knowledge in which full or partial logical omniscience, as discussed in Section 2.2.1, are not assumed.

5.1 Semantics

In the previous chapter, GKSSs were defined as structures where agents' epistemic states were represented by subsets of *OL*, possibly including the $*$ element. Consider a restriction of this semantics to *finite* sets. Let Γ_2 be the following theory:

$$\Gamma_2 = \{\triangle_1\{p\}, \triangle_1\{\triangle_1\{p\}\}, \triangle_1\{\triangle_1\{\triangle_1\{p\}\}\}, \ldots\}$$

Clearly, this theory is not satisfiable by such a semantics, since it describes an agent with an infinite epistemic state. However, like for Γ_1 in Chapter 3 (p. 33), a proof of its inconsistency would require an "infinite" deduction rule and thus an axiomatization such as *EC* without such a rule would be (strongly) incomplete. As another example, consider the following theory:

$$\Gamma_3 = \{\triangle_1\{\alpha, \beta\} \rightarrow \triangle_1\{\alpha \wedge \beta\} : \alpha, \beta \in AL\}$$

Unlike Γ_2, Γ_3 is satisfiable in a semantics of finite epistemic states, but only in a structure in which agent 1's epistemic state is the empty set. Thus,

$$\Gamma_3 \models_f \triangledown_1 \emptyset$$

(where $\models_f$ means logical consequence with respect to the mentioned finite semantics). But again, a proof of $\triangledown_1\emptyset$ from Γ_3 would be infinitely long, and an axiomatization without an infinite deduction rule would thus be (strongly) incomplete since then

$$\Gamma_3 \not\vdash \triangledown_1\emptyset$$

Furthermore, the only plausible semantical explanation of Γ_2 and Γ_3 is that the agents can have infinite epistemic states, so a semantical solution to the problem in the manner of the solution to the problem with Γ_1 in Ch. 3 is not possible. Note that the problems with incompleteness illustrated with Γ_2 and Γ_3 are consequences of the fact that the theories are infinite. Thus only *strong* completeness is sacrificed by using the mentioned semantics; weak completeness and completeness with respect to a class of premises which do cause such problems can still be attained. This is the strategy taken in the following, where the mentioned class is called *finitary theories*.

Formally, finite agents are modelled by *Knowledge Set Structures (KSSs)* which is a restriction of GKSSs to finite states[1].

Definition 5.1 (Knowledge Set Structure) A Knowledge Set Structure (KSS) is an $n+1$-tuple

$$M = (s_1, \ldots, s_n, \pi)$$

where

$$s_i \in \wp^{fin}(OL)$$

and $\pi : \Theta \to \{\mathbf{true}, \mathbf{false}\}$ is a truth assignment. s_i is the epistemic state of agent i, and the set of all epistemic states is $\mathcal{S}^f = \wp^{fin}(OL)$. The set of all GKSSs is denoted $\mathcal{M}_{fin}$. □

Henceforth the notation $\models_f$ is used to refer to satisfiability, or logical consequence, with respect to $\mathcal{M}_{fin}$, and I continue to use the notation $\models$ to refer to satisfiability or consequence with respect to $\mathcal{M}$. Of course, $\mathcal{M}_{fin} \subset \mathcal{M}$. The model class of $\Gamma \subseteq EL$, w.r.t. $\mathcal{M}_{fin}$, is $mod^f(\Gamma) = \{M \in \mathcal{M}_{fin} : M \models_f \Gamma\}$.

Finite agents, as described by KSSs, is the main focus in this thesis. The framework for possibly infinite agents in Chapter 3 will, however, be very useful in this and the next chapter. The logical system *EC* (complete w.r.t. GKSSs) from the previous chapter will still be used with KSSs. The logic over the language *EL* consisting of the logical system *EC* and associated semantics of KSSs will henceforth be called SSEL — *Static Syntactic Epistemic Logic*.

Next, the set of finitary theories, for which *EC* is complete with respect to $\mathcal{M}_{fin}$, is defined.

EC is of course sound with respect to the restricted semantics:

Corollary 5.1

$$\Gamma \vdash \phi \Rightarrow \Gamma \models_f \phi$$ □

PROOF Corollary of Theorem 4.1, since $\mathcal{M}_{fin} \subset \mathcal{M}$. ■

[1] In addition to the infinite states we dispense with the finite states which include the $*$ element. Recall from the discussion in Section 3.6 that we can view these states as infinite, since they cannot be described by any terms.

5.2 Finitary Theories

Definition 5.2 (Finitary Theory) A theory Γ is *finitary* iff it is consistent and for all ϕ,

$$\begin{array}{c}\Gamma \vdash (\triangledown_1 T_1 \wedge \cdots \wedge \triangledown_n T_n) \rightarrow \phi \text{ for all terms } T_1, \ldots, T_n \\ \Downarrow \\ \Gamma \vdash \phi\end{array} \qquad \Box$$

Definition 5.3 (Finitarily Open Theory) A theory Γ is *finitarily open* iff there exist terms $T_1, \ldots, T_n$ such that

$$\Gamma \nvdash \neg(\triangledown_1 T_1 \wedge \cdots \wedge \triangledown_n T_n) \qquad \Box$$

Lemma 5.1

1. A finitary theory is finitarily open.
2. If Γ is a finitary theory and $\Gamma \nvdash \phi$, then $\Gamma \cup \{\neg\phi\}$ is finitarily open. $\Box$

PROOF

1. Let Γ be a finitary theory. If Γ is not finitarily open, $\Gamma \vdash \neg(\triangledown_1 T_1 \wedge \cdots \wedge \triangledown_n T_n)$ for all terms $T_1, \ldots, T_n$. Then, for an arbitrary ϕ, $\Gamma \vdash (\triangledown_1 T_1 \wedge \cdots \wedge \triangledown_i T_n) \rightarrow \phi$ for all $T_1, \ldots, T_n$ and thus $\Gamma \vdash \phi$ since Γ is finitary. By the same argument $\Gamma \vdash \neg\phi$, contradicting the fact that Γ is consistent.
2. Let Γ be a finitary theory, and let $\Gamma \nvdash \phi$. Then there must exist terms $T_1^\phi, \ldots, T_n^\phi$ such that $\Gamma \nvdash (\triangledown_1 T_1^\phi \wedge \cdots \wedge \triangledown_n T_n^\phi) \rightarrow \phi$. By **Prop** we must have that $\Gamma \nvdash \neg\phi \rightarrow \neg(\triangledown_1 T_1^\phi \wedge \cdots \wedge \triangledown_n T_n^\phi)$ and thus that $\Gamma \cup \{\neg\phi\} \nvdash \neg(\triangledown_1 T_1^\phi \wedge \cdots \wedge \triangledown_n T_n^\phi)$, which shows that $\Gamma \cup \{\neg\phi\}$ is finitarily open. ■

Corollary 5.2 Let *Cons*, *FOT* and *Fin* be the set of all consistent, finitarily open and finitary theories, respectively.

$$\mathit{Fin} \subset \mathit{FOT} \subset \mathit{Cons} \qquad \Box$$

PROOF Follows immediately (see Example 5.1 below for examples showing that these subsets are proper.). ■

Example 5.1 The following are examples of non-finitary theories (let $n = 2$ and $p \in \Theta$):

1. $\Gamma_2 = \{\triangle_1\{p\}, \triangle_1\{\triangle_1\{p\}\}, \triangle_1\{\triangle_1\{\triangle_1\{p\}\}\}, \ldots\}$. Γ_2 is not finitarily open, and describes an agent with an infinite epistemic state.
2. $\Gamma_4 = \{\neg \triangledown_1 T : T \in TL\}$. Γ_2 is not finitarily open, and describes an agent which cannot be at any finite point.

3. $\Gamma_5 = \{\triangledown_1 T \rightarrow \neg \triangledown_2 T' : T, T' \in TL\}$. Γ_3 is not finitarily open, and describes a situation where agents 1 and 2 cannot *simultaneously* be at finite points.

4. $\Gamma_6 = \{\triangledown_1 T \rightarrow p : T \in TL\}$. Γ_4 is finitarily open, but not finitary. To see the former, observe that there is no T such that $\Gamma_4 \vdash \neg \triangledown_1 T$ or $\Gamma_4 \vdash \neg \triangledown_2 T$. To see the latter, observe that $\Gamma_4 \nvdash p$ but $\Gamma_4 \vdash (\triangledown_1 T_1 \wedge \triangledown_2 T2) \rightarrow p$ for all T_1, T_2. □

5.3 Completeness

I show that *EC* is complete with respect to $\mathcal{M}_{fin}$, for all finitary theories.

Theorem 5.1 A theory Γ is finitarily open if and only if it is satisfiable in $\mathcal{M}_{fin}$.□

PROOF Γ is finitarily open iff there exist T_i $(1 \le i \le n)$ such that $\Gamma \nvdash \neg(\triangledown_i T_1 \wedge \cdots \wedge \triangledown_n T_n)$; iff, by Cor. 4.3, there exist T_i such that $\Gamma \not\models \neg(\triangledown_i T_1 \wedge \cdots \wedge \triangledown_n T_n)$; iff there exist T_i and a structure $M \in \mathcal{M}$ such that $M \models \Gamma$ and $M \models \triangledown_1 T_1 \wedge \cdots \wedge \triangledown_n T_n$; iff there exist T_i and $M = (s_1, \dots, s_n, \pi) \in \mathcal{M}$ such that $s_i \subseteq [T_i]$ $(1 \le i \le n)$ and $M \models \Gamma$; iff there exist $s_i \in \wp^{fin}(OL)$ $(1 \le i \le n)$ such that $(s_1, \dots, s_n, \pi) \models \Gamma$; iff Γ is satisfiable in $\mathcal{M}_{fin}$. ■

Theorem 5.2 Let $\Gamma \subseteq EL$. $\Gamma \models_f \phi \Rightarrow \Gamma \vdash \phi$ for all ϕ iff Γ is finitary. □

PROOF Let Γ be a finitary theory. In the case that Γ is inconsistent, $\Gamma \vdash \phi$ trivially so let Γ be consistent. Let $\Gamma \models_f \phi$. By Lemma 5.1.1 Γ is finitarily open and thus satisfiable by Theorem 5.1. $\Gamma \cup \{\neg\phi\}$ is unsatisfiable in $\mathcal{M}_{fin}$, and thus not finitarily open. Assume that $\Gamma \nvdash \phi$. Then $\Gamma \cup \{\neg\phi\}$ is finitarily open by Lemma 5.1.2. The contradiction shows that the assumption must be wrong, and that $\Gamma \vdash \phi$.

For the other direction, let $\Gamma \models_f \phi \Rightarrow \Gamma \vdash \phi$ for all ϕ, and assume that $\Gamma \nvdash \phi$. Then, $\Gamma \not\models_f \phi$, that is, there is a $M = (s_1, \dots, s_n, \pi) \in mod^f(\Gamma)$ such that $M \not\models_f \phi$. Let T_i $(1 \le i \le n)$ be terms such that $[T_i] = s_i$. $M \models_f \triangledown_i T_1 \wedge \cdots \wedge \triangledown_n T_n$, and thus $M \not\models_f (\triangledown_1 T_1 \wedge \cdots \wedge \triangledown_n T_n) \rightarrow \phi$. By soundness (Corollary 5.1) $\Gamma \nvdash (\triangledown_1 T_1 \wedge \cdots \wedge \triangledown_n T_n) \rightarrow \phi$, showing that Γ is finitary. ■

5.3.1 Finitaryness of Finite Theories

Theorem 5.2 shows that weak completeness of *EC* corresponds to finitaryness of the empty set. The empty set, and thus every finite set, is indeed finitary[2].

Lemma 5.2 The empty theory is finitary. □

PROOF To save space, the proof is delayed until Chapter 6 where a more general result is shown. See Lemma 6.10 on p. 75. ■

Corollary 5.3 (Weak Completeness) *EC* is weakly complete: $\models_f \phi \Rightarrow \vdash \phi$. □

[2]There does not seem to exist a trivial proof of this.

PROOF Follows immediately from Theorem 5.2 and Lemma 5.2. ■

Corollary 5.4 All finite theories are finitary. □

PROOF Follows immediately from Lemma 5.2 and the following Lemma 5.4 (which does not depend on the current corollary). ■

5.4 Properties of Finitary Theories

Lemma 5.3 Let $\Gamma \subseteq EL$. The following statements are equivalent:

1. Γ is finitary.
2. $\Gamma \models_f \phi \Rightarrow \Gamma \vdash \phi$, for any ϕ
3. $\Gamma \models_f \phi \Rightarrow \Gamma \models \phi$, for any ϕ
4. $(\exists_{M \in mod(\Gamma)} M \models \phi) \Rightarrow (\exists_{M \in mod^f(\Gamma)} M \models_f \phi)$, for any ϕ
5. $\Gamma \nvdash \phi \Rightarrow \Gamma \cup \{\neg\phi\}$ is finitarily open, for any ϕ. □

PROOF

1. $\Leftrightarrow$ 2.: Theorem 5.2.

2. $\Rightarrow$ 3.: If $\Gamma \not\models \phi$ then $\Gamma \nvdash \phi$ by soundness (Th. 4.1), and by 2. above $\Gamma \not\models_f \phi$.

3. $\Rightarrow$ 2.: If $\Gamma \models_f \phi$, then $\Gamma \models \phi$ by 3. above, and $\Gamma \vdash \phi$ by completeness (Th. 4.2).

4. $\Rightarrow$ 2.: Assume that 4. above holds. Let ϕ be a formula, and assume that $\Phi \nvdash \phi$. By Theorem 4.2, $\Phi \not\models \phi$; there exists $M \in mod(\Phi)$ such that $M \not\models \phi$; $M \models \neg\phi$. By 4. above, there exists a $M^f \in mod^f(\Phi)$ such that $M^f \models_f \neg\phi$. Since $M^f \models_f \Phi$ and $M^f \not\models_f \phi$, $\Phi \not\models_f \phi$ showing 2. above.

2. $\Rightarrow$ 4.: Assume that 2. above holds. Let ϕ be a formula, and let $M \in mod(\Phi)$ be such that $M \models \phi$. Since $M \models \Phi$, $\Phi \not\models \neg\phi$ and (by Theorem 4.1) $\Phi \nvdash \neg\phi$. By 2. above, $\Phi \not\models_f \neg\phi$, i.e. there is a $M^f \in mod^f(\Phi)$ such that $M^f \not\models \neg\phi$. Then $M^f \models \phi$, which shows 4. above.

1. $\Rightarrow$ 5.: Lemma 5.1.2.

5. $\Rightarrow$ 4.: Assume that 4. does not hold; that there is a ϕ such that $\Gamma \cup \{\phi\}$ is satisfiable in $\mathcal{M}$ but unsatisfiable in $\mathcal{M}_{fin}$. Since $\Gamma \cup \{\phi\}$ is satisfiable in $\mathcal{M}$, then $\Gamma \not\models \neg\phi$. By soundness, $\Gamma \nvdash \neg\phi$. Since $\Gamma \cup \{\phi\}$ is unsatisfiable in $\mathcal{M}_{fin}$, $\Gamma \cup \{\neg\neg\phi\}$ is not finitarily open by Theorem 5.1. This shows that 5. does not hold (it does not hold for $\neg\phi$). ■

Lemma 5.4 If Γ is a finitary theory and Δ is a finite theory, then $\Gamma \cup \Delta$ is a finitary theory. □

PROOF If $\Gamma \cup \Delta \models_f \phi$ then $\Gamma \models_f \bigwedge\Delta \rightarrow \phi$ (where $\bigwedge\Delta$ is a conjunction of the formulae in Δ). By completeness (Th. 5.2) $\Gamma \vdash \bigwedge\Delta \rightarrow \phi$ and thus $\Gamma \cup \Delta \vdash \phi$. By Th. 5.2 $\Gamma \cup \Delta$ is finitary. ■

5.5 Axiomatization Using Infinite Rules

The reason that the logic *EC* is complete only for finitary theories, is that we cannot axiomatize the fact that certain infinite sets of formulae lead to inconsistency using only finite deduction rules. Here, I describe the logic using *infinite* deduction rules (and thus infinite proofs) and show that it is sound and complete for all theories.

The infinite rule we need is

$$R_f = \frac{\text{for all } T \in TL\text{: } \Gamma \cup \{\triangledown_i T\} \vdash \bot}{\Gamma \vdash \bot}$$

(where $\bot$ is an arbitrary contradiction $\bot = \phi \wedge \neg\phi$).

I now argue briefly that $EC \cup \{R_f\}$ is sound and complete with respect to $\mathcal{M}_{fin}$, for all theories (most of the proof is identical to parts of the two previous soundness and completeness proofs, so I refer to those).

5.5.1 Soundness

For soundness we only have to show that R_f preserves logical consequence. Let $\Gamma \cup \{\triangledown_i T\} \models \bot$ for every term T, and assume that $M \models \Gamma$ where s_i is the epistemic state of i in M. Then $M \not\models \{\triangledown_i T\}$, and thus $s_i \not\subseteq [T]$ for every term T. Since there is a term T for every set $s \in \wp^{fin}(OL)$, this is a contradiction. Thus, Γ has no models and $\Gamma \models \bot$.

5.5.2 Completeness

We can extend every consistent theory Γ to a maximal consistent theory Γ' by using the same construction as in Lemma 4.7. Since Γ is consistent, there is a term T' such that $\Gamma \cup \{\triangledown_i T'\}$ is consistent (since otherwise R_f could be used to show inconsistency of Γ). By construction of Γ', $\triangledown_i T' \in \Gamma'$. Assume that $\Gamma' \vdash \bot$. If this proof is finite, R_f was not applied and we have a contradiction by compactness (see Lemma 4.7). If the proof of $\Gamma' \vdash \bot$ is infinite, then R_f must have been applied. $\Gamma' \cup \{\triangledown_i T'\} \vdash \bot$ must have been among the premises in this application, but this is equal to the conclusion $\Gamma' \vdash \bot$. It is thus clear that we cannot prove the conclusion using R_f, and we cannot have a finite proof either. Thus, Γ' is consistent (and clearly maximal).

Clearly, Lemma 4.8 still holds for $EC \cup \{R_f\}$. The fact that every maximal consistent theory are satisfiable in $\mathcal{M}_{fin}$, can now be shown in exactly the same way as in Lemma 4.9; since the construction of the maximal theory (above) guarantees that $\triangledown_i T' \in \Gamma'$ we can "place" the agent at $[T']$: $s_i^\Gamma = [T']$.

It follows that every consistent theory is satisfiable, and thus that $EC \cup \{R_f\}$ is complete (for all theories) by the same argument as in Theorem 4.2.

5.6 Decidability

In this section, some decidability results are shown. It is assumed that Θ is a recursive set.

Lemma 5.5 Let $T, U \in TL$ be two terms. The question whether $[T] = [U]$ is decidable. $\square$

PROOF $[T] = [U]$ iff $\models T \doteq U$ iff (by completeness of TC) $\vdash T \doteq U$. $[T] \neq [U]$ iff $\models \neg T \doteq U$ iff (by completeness of TC) $\vdash \neg T \doteq U$. To check whether $[T] = [U]$ we can enumerate all proofs in TC, and checking each proof we will eventually come to a proof of either $\vdash T \doteq U$ or $\vdash \neg T \doteq U$. $\blacksquare$

I now show that the satisfaction problem (in $\mathcal{M}_{fin}$) is decidable.

Lemma 5.6 Let $M = (s_1, \dots, s_n, \pi) \in \mathcal{M}_{fin}$ and $\phi \in EL$. The question whether $M \models \phi$ is decidable. $\square$

PROOF The proof is by structural induction over ϕ.

For the first base case, let $\phi = p \in \Theta$. We can then check whether $M \models \phi$ by checking whether $\pi(p) = \mathbf{true}$.

For the second base case, let $\phi = T \doteq U$. $M \models \phi$ iff $[T] = [U]$ which is decidable (Lemma 5.5).

For the third base case, let $\phi = \triangle_i T$. $M \models \phi$ iff $[T] \subseteq s_i$. $s_i \subset OL$ is a finite set, and it is easy to see that we can find a term $U \in TL$ such that $[U] = s_i$ in finite time (write down the set s_i as a term by choosing an arbitrary ordering of the elements in the set, and do this recursively for elements containing new sets . The number of nestings is finite, by construction of OL.). As argued in the second base case, we can decide whether $[T \sqcup U] = [U]$, i.e. whether $[T] \subseteq [U]$, i.e. whether $[T] \subseteq s_i$. For the fourth base case: $\phi = \triangledown_i T$, $M \models \phi$ iff $s_i \subseteq [T]$, which we can decide in a similar way to the third base case.

For the first inductive step, let $\phi = \neg\psi$. $M \models \phi$ iff $M \not\models \psi$ which is decidable by the induction hypothesis. Similarly for the second inductive step: $\phi = (\psi_1 \wedge \psi_2)$. $M \models \phi$ iff $M \models \psi_1$ and $M \models \psi_2$ which is decidable by the induction hypothesis.

Thus, $M \models \phi$ is decidable for all ϕ. $\blacksquare$

Lemma 5.7 If Γ is finitary and $\Gamma \models_f \phi$, then there exists a decision procedure that determines the fact that $\Gamma \models_f \phi$. $\square$

PROOF Since $\Gamma \models_f \phi$, $\Gamma \vdash \phi$ by completeness since Γ is finitary. We can enumerate all proofs $\Gamma \vdash \phi'$, and sooner or later we will come to one where $\phi' = \phi$. By soundness, this procedure will only answer "yes" if indeed $\Gamma \models_f \phi$. $\blacksquare$

Lemma 5.8 If $\Gamma \subseteq EL$, $mod^f(\Gamma)$ is recursive and $\Gamma \not\models_f \phi$, then there exists a decision procedure that determines the fact that $\Gamma \not\models_f \phi$. $\square$

PROOF If $\Gamma \not\models_f \phi$ then there is a $M \in mod^f(\Gamma)$ such that $M \not\models \phi$. We can enumerate all $M' \in mod^f(\Gamma)$, and determine whether $M' \models \phi$ (Lemma 5.6), and sooner or later we will find an M such that $M \not\models \phi$. This procedure will only answer "yes" if indeed $\Gamma \not\models_f \phi$. $\blacksquare$

Theorem 5.3 Let Γ be a finitary theory such that $mod^f(\Gamma)$ is recursive. The problem of logical consequence from Γ w.r.t. $\mathcal{M}_{fin}$, and the problem of provability from Γ, are decidable. $\square$

PROOF By Lemmas 5.7 and 5.8, there are procedures that will answer "yes" if $\Gamma \models_f \phi$ or $\Gamma \not\models_f \phi$ respectively. If we run these procedures simultaneously or, equivalently, alternate between checking a proof and a Γ-model, one of the procedures will sooner or later answer "yes". ∎

Corollary 5.5 The problem of validity in $\mathcal{M}_{fin}$, and the problem of theoremhood in *EC*, are decidable. □

PROOF $mod^f(\emptyset) = \mathcal{M}_{fin}$ is recursive since 1) *OL* is recursive by definition, 2) the sets of finite subsets of a recursive set is recursive, 3) the set of all truth assignments is recursive when Θ is, and 4) the Cartesian product of recursive sets is recursive.

Thus, since $\emptyset$ is finitary (Lemma 5.2), the corollary follows from Theorem 5.3. ∎

Chapter 6

Extensions

6.1 Introduction

Epistemic properties of the individual agents can be modeled by imposing constraints on the epistemic states. Ideally, such semantic constraints should correspond to the acceptance of certain axioms. In modal epistemic logic, the correspondence between axioms and semantical constraints has been extensively studied.

In this chapter, I discuss extending the calculus *EC* with additional axioms Φ. I first discuss how to construct the model classes $mod(\Phi) = \{M \in \mathcal{M} : M \models \Phi\}$ and $mod^f(\Phi) = \{M \in \mathcal{M}_{fin} : M \models \Phi\}$ in general, before I investigate the case when Φ describes solely epistemic properties – I call such Φ *epistemic axioms*. I show that for epistemic axioms, the corresponding model class can be obtained by removing illegal epistemic states locally for each agent. In Section 6.6 algebraic conditions on the sets of legal epistemic states which guarantees that the axioms are finitary are developed. These conditions enables extension of *EC*: if the axioms are finitary, then *EC* extended with the axioms is complete with respect to the class of models with legal epistemic states. Examples of extensions with well known epistemic axioms are presented in Section 6.7.

Agents are still assumed to be finite, as described by KSSs $\mathcal{M}_{fin}$ with possible epistemic states $\mathcal{S}^f$ (Def 5.1). However, the more general semantics, GKSSs $\mathcal{M}$ with possible epistemic states $\mathcal{S}$ (Def. 3.8), will be very useful in this chapter. Henceforth the superscript f will be used to denote the finite restriction of a set $\mathcal{M}' \subseteq \mathcal{M}$ or a set $S \subseteq \mathcal{S}$: $\mathcal{M}'^f = \mathcal{M}' \cap \mathcal{M}_{fin}$ and $S^f = S \cap \mathcal{S}^f$.

6.2 Axioms and Model Class Construction

Given a set of axioms Φ, we will construct a class of structures $\mathcal{M}^\Phi$ such that

$$\mathcal{M}^\Phi = mod(\Phi)$$

First, the model class $\mathcal{M}^\phi$ for a single formula ϕ is constructed:

Definition 6.1 ($\mathcal{M}^\phi$) For each formula $\phi \in EL$, $\mathcal{M}^\phi \subseteq \mathcal{M}$ is defined by structural induction over ϕ as follows:

$$\begin{aligned}
\phi = p \in \Theta : &\quad \mathcal{M}^\phi = \{(s_1,\ldots,s_n,\pi) \in \mathcal{M} : \pi(p) = \mathbf{true}\} \\
\phi = \triangle_i T : &\quad \mathcal{M}^\phi = \{(s_1,\ldots,s_n,\pi) \in \mathcal{M} : [T] \subseteq s_i\} \\
\phi = \triangledown_i T : &\quad \mathcal{M}^\phi = \{(s_1,\ldots,s_n,\pi) \in \mathcal{M} : s_i \subseteq [T]\} \\
\phi = \neg\psi : &\quad \mathcal{M}^\phi = \mathcal{M} \backslash \mathcal{M}^\psi \\
\phi = \psi_1 \wedge \psi_2 : &\quad \mathcal{M}^\phi = \mathcal{M}^{\psi_1} \cap \mathcal{M}^{\psi_2}
\end{aligned} \quad \square$$

It is easy to see that the corresponding construction of model classes of formulae including the derived operators is done by taking:

$$\begin{aligned}
\phi = \Diamond_i T : &\quad \mathcal{M}^\phi = \{(s_1,\ldots,s_n,\pi) \in \mathcal{M} : s_i = [T]\} \\
\phi = \psi_1 \vee \psi_2 : &\quad \mathcal{M}^\phi = \mathcal{M}^{\psi_1} \cup \mathcal{M}^{\psi_2} \\
\phi = \psi_1 \rightarrow \psi_2 = \neg\psi_1 \vee \psi_2 : &\quad \mathcal{M}^\phi = \mathcal{M} \backslash \mathcal{M}^{\psi_1} \cup \mathcal{M}^{\psi_2}
\end{aligned}$$

The construction in Def. 6.1 is a straightforward application of the rules of the semantics, and it is trivial to show that $\mathcal{M}^\phi$ in fact is the model class of ϕ:

Lemma 6.1

$$\mathcal{M}^\phi = mod(\phi) \quad \square$$

PROOF I show that

$$M \models \phi \Leftrightarrow M \in \mathcal{M}^\phi$$

where $M = (s_1,\ldots,s_n,\pi)$ by structural induction over ϕ:

$\phi = p \in \Theta$: $M \models \phi \Leftrightarrow \pi(p) = \mathbf{true} \Leftrightarrow M \in \mathcal{M}^\phi$

$\phi = \triangle_i T$: $M \models \phi \Leftrightarrow [T] \subseteq s_i \Leftrightarrow M \in \mathcal{M}^\phi$

$\phi = \triangledown_i T$: $M \models \phi \Leftrightarrow s_i \subseteq [T] \Leftrightarrow M \in \mathcal{M}^\phi$

$\phi = \neg\psi$: By the induction hypothesis, $M \models \psi \Leftrightarrow M \in \mathcal{M}^\psi$. $M \models \phi \Leftrightarrow M \not\models \psi \Leftrightarrow M \notin \mathcal{M}^\psi \Leftrightarrow M \in \mathcal{M}^\phi$.

$\phi = \psi_1 \wedge \psi_2$: By the induction hypothesis, $M \models \psi_1 \Leftrightarrow M \in \mathcal{M}^{\psi_1}$ and $M \models \psi_2 \Leftrightarrow M \in \mathcal{M}^{\psi_2}$. $M \models \phi \Leftrightarrow M \models \psi_1$ and $M \models \psi_2 \Leftrightarrow M \in \mathcal{M}^{\psi_1}$ and $M \in \mathcal{M}^{\psi_2} \Leftrightarrow M \in \mathcal{M}^\phi$. ■

The model class of a set of formulae Φ is, of course, the intersection of the model classes for the individual formulae:

$$\mathcal{M}^\Phi = (\bigcap_{\phi \in \Phi} \mathcal{M}^\phi) \cap \mathcal{M} \tag{6.1}$$

Note that $\mathcal{M}^\emptyset = \mathcal{M}$.

$\mathcal{M}^\Phi$ is all the models of Φ in $\mathcal{M}$, where the agents are not necessarily restricted to finite points. Recall that the models of Φ in $\mathcal{M}_{fin}$, where the agents are restricted to finite points, is denoted $mod^f(\Phi)$. Clearly,

$$\mathcal{M}^\Phi = mod(\Phi)$$

and

$$\mathcal{M}^{\Phi f} = mod^f(\Phi)$$

When constructing the model classes, the usual care must be taken to specify whether we want to describe *an axiom* (a formula) or an *axiom schema* (a set of formulae). As a sentence which can be interpreted as both, consider

$$\triangle_i T \to \triangle_i U$$

When this sentence is specified as an axiom, we should construct the model class $\mathcal{M}^{\phi}$ taking the unspecified sets T, U and the agent name i as parameters:

$$M_i^{\phi} = \mathcal{M} \backslash \{(s_1, \ldots, s_n, \pi) : [T] \subseteq s_i\} \cup \{(s_1, \ldots, s_n, \pi) : [U] \subseteq s_i\}$$

When specified as an axiom schema, we should try to construct the model class $\mathcal{M}^{\Phi}$ for the set Φ of all formulae satisfying the schema. In the current case,

$$\mathcal{M}^{\Phi} = \emptyset$$

(Of course, we may want to fix only *some* of the parameters in an axiom schema).

6.3 Epistemic Axioms

The most probable reason for extending the system *EC* by adding new axioms — or, equivalently, impose semantical constraints by restricting the set of allowed structures — is that we want to model certain epistemic properties of the agents, e.g. the fact that they will never "know" a contradiction. Not all formulae in *EL* should be considered as candidates for describing epistemic properties. One example is $p \to \triangle_i\{p\}$. This formula does not solely describe the *agent* – it describes a relationship between the agent and the world, i.e. about an agent in a situation. Another example is $\Diamond_i\{p\} \to \Diamond_j\{q\}$, which describes a constraint on one agent's belief set contingent on another agent's belief set. Neither of these two formulae describe purely *epistemic* properties of an agent. Candidates for *epistemic axioms* should therefore a) only refer to epistemic facts and not to external facts and b) only describe one particular agent. We will call formulae satisfying a) *epistemic* formulae. In the following definition, *EF* is the set of epistemic formulae and *Ax* is the set of candidate epistemic axioms.

Definition 6.2 (*EF*, *EF^i*, *Ax*)

- $EF \subseteq EL$ is the least set such that

$$\begin{array}{rll} \text{If } T \in TL \text{ then} & \triangle_i T, \bigtriangledown_i T \in EF & (1 \leq i \leq n) \\ \text{If } \phi, \psi \in EF \text{ then} & \neg\phi, (\phi \wedge \psi) \in EF & \end{array}$$

- $EF^i = \{\phi \in EF : \text{Every epistemic operator in } \phi \text{ is an } i\text{-op.}\}$ $(1 \leq i \leq n)$
- $Ax = \bigcup_{1 \leq i \leq n} EF^i$ □

An example of an epistemic axiom schema is

$$\triangle_i\{\alpha \wedge \beta\} \to \triangle_i\{\alpha\} \wedge \triangle_i\{\beta\}$$

6.3.1 Model Class Construction for Epistemic Axioms

I now show that, in the case that $\phi \in Ax$, we can rephrase Def. 6.1 in terms of sets of legal epistemic states for each agent.

Definition 6.3 ($\mathcal{M}^{\phi}_{Ax}$, S^{ϕ}_i) For each epistemic formula $\phi \in EF^i$,

$$\mathcal{M}^{\phi}_{Ax} = \{(s^{\phi}_1, \ldots, s^{\phi}_n, \pi) \in \mathcal{M} : s_i \in S^{\phi}_i\}$$

where S^{ϕ}_i is constructed by structural induction over ϕ as follows:

$$\begin{aligned}
\phi = \triangle_i T : &\quad S^{\phi}_i = \{X \in \mathcal{S} : [T] \subseteq X\} \\
\phi = \triangledown_i T : &\quad S^{\phi}_i = \{X \in \mathcal{S} : X \subseteq [T]\} \\
\phi = \neg\psi : &\quad S^{\phi}_i = \mathcal{S} \backslash S^{\psi}_i \\
\phi = \psi_1 \wedge \psi_2 : &\quad S^{\phi}_i = S^{\psi_1}_i \cap S^{\psi_2}_i
\end{aligned} \quad \square$$

In the construction of $\mathcal{M}^{\phi}_{Ax}$ we remove the impossible epistemic states by restricting the set of epistemic states to S^{ϕ}_i. The epistemic states which are not removed are the possible states — an agent can be placed in any of these states and will satisfy the epistemic axiom ϕ.

Again, it is easy to see that the corresponding construction of model classes of formulae including the derived operators is done by taking:

$$\begin{aligned}
\phi = \Diamond_i T : &\quad S^{\phi}_i = \{[T]\} \\
\phi = \psi_1 \vee \psi_2 : &\quad S^{\phi}_i = S^{\psi_1}_i \cup S^{\psi_2}_i \\
\phi = \psi_1 \rightarrow \psi_2 = \neg\psi_1 \vee \psi_2 : &\quad S^{\phi}_i = \mathcal{S} \backslash S^{\psi_1}_i \cup S^{\psi_2}_i
\end{aligned}$$

Definition 6.4 ($\mathcal{M}^{\Phi}_{Ax}$, S^{Φ}_i) Let $\Phi \subseteq Ax$.

$$\begin{aligned}
S^{\Phi}_i &= \left(\textstyle\bigcap_{\phi \in \Phi \cap EF^i} S^{\phi}_i\right) \cap \mathcal{S} \\
\mathcal{M}^{\Phi}_{Ax} &= \{(s^{\Phi}_1, \ldots, s^{\Phi}_n, \pi) \in \mathcal{M} : s_i \in S^{\Phi}_i\}
\end{aligned} \quad \square$$

Lemma 6.2

$$\mathcal{M}^{\Phi}_{Ax} = mod(\Phi) \quad \square$$

PROOF Let ϕ be a formula. I first show that

$$M \models \phi \Leftrightarrow M \in \mathcal{M}^{\phi}_{Ax} \tag{6.2}$$

where $M = (s_1, \ldots, s_n, \pi)$ by structural induction over $\phi \in EF^i$:

$\phi = \triangle_i T$: $M \models \phi \Leftrightarrow [T] \subseteq s_i \Leftrightarrow s_i \in S^{\phi}_i \Leftrightarrow M \in \mathcal{M}^{\phi}_{Ax}$

$\phi = \triangledown_i T$: $M \models \phi \Leftrightarrow s_i \subseteq [T] \Leftrightarrow s_i \in S^{\phi}_i \Leftrightarrow M \in \mathcal{M}^{\phi}_{Ax}$

$\phi = \neg\psi$: By the induction hypothesis, $M \models \psi \Leftrightarrow M \in \mathcal{M}^{\psi}_{Ax}$. $M \models \phi \Leftrightarrow M \not\models \psi \Leftrightarrow M \notin \mathcal{M}^{\psi}_{Ax} \Leftrightarrow s_i \notin S^{\psi}_i \Leftrightarrow s_i \in \mathcal{S} \backslash S^{\psi}_i = S^{\phi}_i \Leftrightarrow M \in \mathcal{M}^{\phi}_{Ax}$.

$\phi = \psi_1 \wedge \psi_2$: By the induction hypothesis, $M \models \psi_1 \Leftrightarrow M \in \mathcal{M}_{Ax}^{\psi_1}$ and $M \models \psi_2 \Leftrightarrow M \in \mathcal{M}_{Ax}^{\psi_2}$. $M \models \phi \Leftrightarrow M \models \psi_1$ and $M \models \psi_2 \Leftrightarrow M \in \mathcal{M}_{Ax}^{\psi_1}$ and $M \in \mathcal{M}_{Ax}^{\psi_2} \Leftrightarrow s_i \in S_i^{\psi_1}$ and $s_i \in S_i^{\psi_2} \Leftrightarrow s_i \in S_i^{\psi_1} \cap S_i^{\psi_2} = S_i^{\phi} \Leftrightarrow M \in \mathcal{M}_{Ax}^{\phi}$.

Clearly, $\mathcal{M}_{Ax}^{\Phi} = (\bigcap_{\phi \in \Phi} \mathcal{M}_{Ax}^{\phi}) \cap \mathcal{M}$, which is equal to $(\bigcap_{\phi \in \Phi} mod(\phi)) \cap \mathcal{M}$ by (6.2) which is equal to $mod(\Phi)$. ∎

Note that $S_i^{\emptyset} = \mathcal{S}$ and, again, $\mathcal{M}_{Ax}^{\emptyset} = \mathcal{M}$.

The corresponding model class in $\mathcal{M}_{fin}$ is

$$\mathcal{M}_{Ax}^{\Phi\, f} = \mathcal{M}_{Ax}^{\Phi} \cap \mathcal{M}_{fin} = \{(s_1^{\Phi}, \ldots, s_n^{\Phi}, \pi) : s_i^{\Phi} \in S_i^{\Phi f}\} \tag{6.3}$$

where $S_i^{\Phi f} = S_i^{\Phi} \cap \mathcal{S}^f$. Clearly,

$$\mathcal{M}_{Ax}^{\Phi\, f} = mod^f(\Phi)$$

Thus, the model class for epistemic axioms is constructed by removing certain states from the set of legal epistemic states. For example, the schema

$$\triangle_i\{\alpha \wedge \beta\} \rightarrow \triangle_i\{\alpha\} \wedge \triangle_i\{\beta\} \tag{6.4}$$

corresponds to removing epistemic states where the agent knows a conjunction without knowing the conjuncts. This would be the case if the agent had a reasoning mechanism which would never produce a conjunction without simultaneously producing the conjuncts.

6.4 Extensions: Soundness and Completeness

In Section 5.3 it was shown that *EC* is complete with respect to $\mathcal{M}_{fin}$ for finitary premises; i.e. $\Gamma \models_f \phi \Rightarrow \Gamma \vdash \phi$ iff Γ is finitary. The symbol $\vdash_{\Phi}$ is now used to denote derivability in *EC* extended by axioms Φ, and $\models_{\Phi,f}$ is used to denote satisfiability/logical consequence in $mod^f(\Phi)$.

The new system is sound and complete with respect to $mod^f(\Phi)$, for theories Γ such that $\Phi \cup \Gamma$ is finitary. Particularly, the new system is weakly complete iff Θ is finitary:

Lemma 6.3 $\vdash_{\Phi} \phi \Leftrightarrow \models_{\Phi,f} \phi$ for all ϕ iff Φ is finitary. □

PROOF Follows directly from Theorem 5.2 and Corollary 5.1. ∎

Thus, if finitaryness of an axiom schema can be shown, the extension of *EC* is sound and (weakly) complete with respect to the corresponding model class. This is similar to the situation in modal logic, where the basic (normal) system can be extended with certain axiom schemata to obtain (weakly) complete systems with respect to the corresponding model classes.

A particularly interesting case is epistemic axioms. It was shown in the previous section that the model classes for epistemic axioms can be constructed removing certain states from the set of legal epistemic states locally for each agent. A system with such a set of axioms added can be used to model agents with reasoning mechanisms which never can produce certain states. Note that the adoption of epistemic axioms such as (6.4) does not entail full or partial logical omniscience (Section 2.2.1); they can be explained in terms of a reasoning mechanism avoiding certain states.

The problem of extension requires a method for deciding finitaryness of axioms. It turns out that finitaryness is non-trivial to prove. In the next sections, sufficient finitaryness conditions for the case of epistemic axioms are developed. Examples of extensions with well known axioms are shown.

6.5 Finitary Structures

The definition of a finitary theory is a syntactic one. Here, a similar semantic concept is introduced.

I define a notion of finitariness for classes of structures, which is a kind of a "finite model property".

Definition 6.5 (Finitary set of GKSSs) A class of general knowledge set structures $\mathcal{M}' \subseteq \mathcal{M}$ is finitary iff, for all ϕ:

$$\exists_{M \in \mathcal{M}'} M \models \phi$$
$$\Downarrow$$
$$\exists_{M^f \in \mathcal{M}'^f} M^f \models \phi \qquad \Box$$

Lemma 6.4 Let $\Gamma \subseteq EL$. Γ is finitary iff $mod(\Gamma)$ is finitary. $\Box$

PROOF Follows directly from Def. 6.5 and Lemma 5.3. ■

6.6 Semantic Finitaryness Conditions for Epistemic Axioms

In this section, algebraic conditions on sets of epistemic states are defined. These conditions can be used to show whether a set of epistemic axioms is finitary. First, two general algebraic conditions on sets are defined.

Definition 6.6 (Directed Set) A set A with a reflexive and transitive relation $\leq$ is *directed* iff for every finite subset B of A, there is an element $a \in A$ such that $b \leq a$ for every $b \in B$. $\Box$

When I in the following refer to a set of sets as directed, I implicitly mean with respect to subset inclusion.

Definition 6.7 (Cover) A family of subsets of a set A whose union includes A is a *cover* of A. $\Box$

If B is a cover of A, then $\cup B$ *covers* A.

The following condition states a collection of conditions on sets of epistemic states. The main result at the end of this section is that these conditions are sufficient for the GKSSs spanned out by the sets of epistemic states to be finitary (as defined in the previous section), and furthermore, if the sets are induced by epistemic axioms (as defined in Section 6.3), that the axioms are finitary.

Definition 6.8 (Finitary Set of Epistemic States) If $S \subseteq \mathcal{S}$ is a set of epistemic states and $s \in \wp(OL)$, then the set of finite subsets of s included in S is denoted

$$S|_s^f = S \cap \wp^{fin}(s)$$

S is *finitary* iff:

1. For every infinite $s \in S$:
   (a) $S|_s^f$ is directed
   (b) $S|_s^f$ is a cover of s
2. $\forall_{s \cup \{*\} \in S} \forall_{s' \in \wp^{fin}(OL)} \exists_{\alpha \notin s'}$:
   (a) $\exists_{s^f \in S \cap \wp(s \cup \{\alpha\})} s' \cap s \subseteq s^f$
   (b) $\exists_{s^f \in S \cap \wp(s \cup \{\alpha\})} s^f \not\subseteq s'$
   (c) $S \cap \wp(s \cup \{\alpha\})$ is directed □

Now, some properties of sets of finitary epistemic states are shown. Recall that S^f is used to denote the subset of *finite* epistemic states (without the $*$ element) in a set of epistemic states S.

Lemma 6.5 Let $S \subseteq \mathcal{S}$ be a set of epistemic states. Definition 6.8.1 holds iff for every infinite $s \in S$

$$\forall_{s' \in \wp^{fin}(s)} \exists_{s^f \in S^f} s' \subseteq s^f \subseteq s \tag{6.5}$$

□

PROOF

⇒) Assume that Def. 6.8.1 holds, and let $s' \in \wp^{fin}(s)$. s' is finite, say $s' = \{\beta_1, \ldots, \beta_k\}$. Because $s' \subseteq s$, by Def. 6.8.1.b) $s' \subseteq \bigcup(S \cap \wp^{fin}(s))$, so for each β_j there is a $t_j \in (S \cap \wp^{fin}(s))$ such that $\beta_j \in t_j$. By Def. 6.8.1.a), there is a $s^f \in (S \cap \wp^{fin}(s))$ such that $\bigcup_{1 \leq j \leq k}\{t_j\} \subseteq s^f$. Then $s' \subseteq s^f$. Since $s^f \in (S \cap \wp^{fin}(s))$, $s' \in S^f$ and $s^f \subseteq s$.

⇐) Assume that (6.5) holds.

Def. 6.8.1.a) Let S' be a finite subset of $S \cap \wp^{fin}(s)$. Clearly, $s' = \bigcup S' \in \wp^{fin}(s)$, and by (6.5) there is a $s^f \in S \cap \wp^{fin}(s)$ such that $s' \subseteq s^f$.

Def. 6.8.1.b) Let $\alpha \in s$. Because $\{\alpha\} \in \wp^{fin}(s)$, by (6.5) there is a $s^f \in S \cap \wp^{fin}(s)$ such that $\{\alpha\} \in s^f$. Then, $\alpha \in \bigcup(S \cap \wp^{fin}(s))$. Thus, $s \subseteq \bigcup(S \cap \wp^{fin}(s))$, and Def. 6.8.1.b holds. ■

Lemma 6.6 Let $S \subseteq \mathcal{S}$ be a set of epistemic states. If S is finitary then:

1. For every infinite $s \in S$:
   (a) $\exists_{s^f \in S^f} s^f \subseteq s$
   (b) $\forall_{s' \in \wp^{fin}(OL)} \exists_{s^f \in S^f} (s^f \subseteq s \text{ and } s^f \not\subseteq s')$

2. $\forall_{s\cup\{*\}\in S}\forall_{s'\in\wp^{fin}(OL)}\exists_{\alpha\notin s'}$:
   (a) $\forall_{s''\subseteq s'} s'' \subseteq s \Rightarrow \exists_{s^f\in S\cap\wp(s\cup\{\alpha\})} s'' \subseteq s^f$
   (b) $\forall_{s''\subseteq s'}\exists_{s^f\in S\cap\wp(s\cup\{\alpha\})} s^f \not\subseteq s''$
   (c) $S\cap\wp(s\cup\{\alpha\})$ is directed □

PROOF 1. Let $s \in S$ be infinite.

(a) Follows from Lemma 6.5 by letting $s' = \emptyset$.

(b) Assume that 1b) does *not* hold for s, i.e. that

$$\exists_{s'\in\wp^{fin}(OL)}\forall_{s^f\in S^f}(s^f \subseteq s \Rightarrow s^f \subseteq s')$$

That is, there is a $s' \in \wp^{fin}(OL)$ such that

$$\forall_{s^f\in S\cap\wp^{fin}(s)} s^f \subseteq s'$$

in other worlds

$$(\bigcup(S\cap\wp^{fin}(s))) \subseteq s'$$

Since s is infinite and s' is finite, $s \not\subseteq s'$ and thus

$$s \not\subseteq (\bigcup(S\cap\wp^{fin}(s)))$$

which contradicts the fact that $\bigcup(S\cap\wp^{fin}(s))$ covers s. Thus, 1b) must hold.

2. Let $s\cup\{*\} \in S$ and $s' \in \wp^{fin}(OL)$, and let α be as defined in Def. 6.8.2.

(a) Let $s'' \subseteq s'$ and $s'' \subseteq s$. By Def. 6.8.2.a there is a $s^f \in S\cap\wp(s\cup\{\alpha\})$ such that $s'\cap s \subseteq s^f$. Since $s'' \subseteq s'\cap s$, $s'' \subseteq s^f$ which proves 2a).

(b) Let $s'' \subseteq s'$. By Def. 6.8.2.b there is a $s^f \in S\cap\wp(s\cup\{\alpha\})$ such that $s^f \not\subseteq s'$. Then, $s^f \not\subseteq s''$ which proves 2b).

(c) Def. 6.8.2.c. ■

The following definitions and intermediate results are needed in the main proof of Lemma 6.8. When ϕ, ψ are formulae, I use the notation $\psi \leq \phi$ to denote that ψ is a (not necessarily proper) subformula of ϕ. By "subformula" I mean that ψ occurs *outside a term* in ϕ; I do not view AL-formulae inside terms in ϕ as subformulae of ϕ (for example, $\phi = \triangle_1\{\triangledown_2\{p\}\} \wedge q$ has two proper subformulae – $\triangle_1\{\triangledown_2\{p\}\}$ and q, but $\triangledown_2\{p\}$ or p are not subformulae of ϕ).

Definition 6.9 (s_i^ϕ) Given a formula $\phi \in EL$,

$$s_i^\phi = \bigcup_{\triangle_i T\leq\phi \text{ or } \triangledown_i T\leq\phi} [T]$$

for each $1 \leq i \leq n$. □

Observe that s_i^ϕ is always finite, since $[T]$ is finite for any term and a formula only has a finite number of subformulae.

Lemma 6.7 If i is an agent, $\phi \in EL$, $S \subseteq \mathcal{S}$ a finitary set of epistemic states and $s = \hat{s} \cup \{*\} \in S$, then there exists a

$$\alpha_{i,S}^{\phi,s} \notin s_i^{\phi} \tag{6.6}$$

where s_i^{ϕ} is defined in Def. 6.9, such that

1. $\forall_{s'' \subseteq s_i^{\phi}} s'' \subseteq \hat{s} \Rightarrow \exists_{s_i^f \in S \cap \wp(\hat{s} \cup \{\alpha_{i,S}^{\phi,s}\})} s'' \subseteq s_i^f$
2. $\forall_{s'' \subseteq s_i^{\phi}} \exists_{s_i^f \in S \cap \wp(\hat{s} \cup \{\alpha_{i,S}^{\phi,s}\})} s_i^f \not\subseteq s''$
3. $S \cap \wp^{fin}(\hat{s} \cup \{\alpha_{i,S}^{\phi,s}\})$ is directed

Note that, given i, S, s and ϕ, there may exist more than one $\alpha \in OL$ satisfying (6.6) and the tree properties above, but we select one of them (arbitrarily) and call it $\alpha_{i,S}^{\phi,s}$. □

PROOF Follows from Lemma 6.6.2, since S is finitary, $\hat{s} \cup \{*\} \in S$ and $s_i^{\phi} \in \wp^{fin}(OL)$. ■

Definition 6.10 ($\tilde{s}_{i,S}^{\phi,s}$) Let i be an agent, $\phi \in EL$, S a finitary set of epistemic states and $s \in S$.

Let

$$\tilde{s}_{i,S}^{\phi,s} = \begin{cases} s & \text{if } * \notin s \\ (s \setminus \{*\}) \cup \{\alpha_{i,S}^{\phi,s}\} & \text{if } * \in s \end{cases}$$ □

($\tilde{s}_{i,S}^{\phi,s}$ is s possibly with the asterisk replaced by $\alpha_{i,S}^{\phi,s}$).

Lemma 6.8 If $S_1, \ldots, S_n$ are finitary sets of epistemic states (Def. 6.8), then

$$\{(s_1, \ldots, s_n, \pi) : s_i \in S_i, \pi \in \Pi\}$$

where Π is all truth assignments, is a finitary set of GKSSs (Def. 6.5). □

PROOF Let $S_1, \ldots, S_n$ be finitary sets of epistemic states. Let ϕ be an arbitrary formula and let $(s_1, \ldots, s_n, \pi) \in S_1 \times \cdots \times S_n \times \Pi$ be such that

$$(s_1, \ldots, s_n, \pi) \models \phi \tag{6.7}$$

I show that there are $s_1^f, \ldots, s_n^f \in S_1^f \times \cdots \times S_n^f$ such that

$$(s_1^f, \ldots, s_n^f, \pi) \models \phi \tag{6.8}$$

For each s_i, either $* \in s_i$ or $* \notin s_i$. When $* \in s_i$, the following shorthand notation is used:

$$\alpha_i = \alpha_{i,S_i}^{\phi,s_i}$$

where $\alpha_{i,S_i}^{\phi,s_i}$ is defined in Lemma 6.7, and

$$\alpha_i \notin s_i^{\phi}$$

Similarly, taking $S = S_i$ and $s = s_i$ in Def. 6.10, the following shorthand notation is used:

$$\tilde{s_i} = \tilde{s_i}_{i,S_i}^{\phi,s_i}$$

If $\psi \in EL$, let $L(\psi)$ be the following statement

$$L(\psi) : \exists_{s_1^f,\ldots,s_n^f} \begin{cases} \text{a) } s_i^f \in S_i^f \\ \text{b) } s_i^f \subseteq \tilde{s_i} \\ \text{c) } (s_1',\ldots,s_n',\pi) \models \psi \text{ for all } s_i' \text{ s. t. } s_i^f \subseteq s_i' \subseteq \tilde{s_i} \end{cases}$$

and let $P(\psi)$ be the following statement

$$P(\psi) : \psi \leq \phi \Rightarrow \begin{cases} 1)\ (s_1,\ldots,s_n,\pi) \models \psi \Rightarrow\ L(\psi) \\ 2)\ (s_1,\ldots,s_n,\pi) \models \neg\psi \Rightarrow\ L(\neg\psi) \end{cases}$$

If $P(\phi)$ holds, then, since $\phi \leq \phi$, $L(\phi)$ holds by 1) and (6.7). $P(\phi)$ is a stronger statement than the Lemma, and is needed for the inductive structure of the proof. By taking $s_i' = s_i^f$ in c) (in $L(\phi)$), we get that $(s_1^f,\ldots,s_n^f,\pi) \models \phi$ for $s_i^f \in S_i^f$, which proves the Lemma.

Before the main proof of $P(\phi)$, one property of S_i is shown: for every $i \in [1,n]$:

$$\exists_{s_i^f \in S_i^f} s_i^f \subseteq \tilde{s_i} \tag{6.9}$$

To see that (6.9) holds, first consider the case that s_i is finite. If $* \notin s_i$, then $s_i^f = s_i \in S_i^f$ and $s_i^f \subseteq \tilde{s_i} = s_i$. If $* \in s_i$, then there is a $s_i^f \subseteq \tilde{s_i}$ by Lemma 6.7.2 (with $S = S_i$ and $s = s_i$, take e.g. $s'' = s_i^\phi$) and $s_i^f \in S_i^f$ since $\tilde{s_i}$ is finite. Second, in the case that s_i is infinite then $\tilde{s_i} = s_i$ and (6.9) holds by Lemma 6.6.1.a.

I now prove $P(\psi)$ for all formulae $\psi \leq \phi$ (including ϕ), by induction over the structure of ψ [1].

$\psi = p \in \Theta$:

1. Assume that $p \leq \phi$, and that $(s_1,\ldots,s_n,\pi) \models p$. I show $L(p)$. For each i there is, by (6.9), a $s_i^f \in S_i^f$ such that $s_i^f \subseteq \tilde{s_i}$, satisfying a) and b) for each i. c) holds since $\pi(p) =$ **true**.
2. Substitute $\neg p$ for p in the above argument.

$\psi = T \doteq U$:

1. Assume that $T \doteq U \leq \phi$, and that $(s_1,\ldots,s_n,\pi) \models T \doteq U$. We show $L(T \doteq U)$. There exists s_i^f satisfying a) and b) by the same argument as in the $\psi = p$ case, and c) holds since $[T] = [U]$.
2. Substitute $\neg T \doteq U$ for $T \doteq U$ and $[T] \neq [U]$ for $[T] = [U]$ in the above argument.

$\psi = \triangle_i T$: Assume that $\triangle_i T \leq \phi$. In both the following cases, let s_j^f, $j \neq i$, be such that $s_j^f \in S_j^f$ and $s_j^f \subseteq \tilde{s_j}$ (giving a) and b) for $j \neq i$) – existence of such s_j^f is given by (6.9) – and let s_j' be arbitrary such that $s_j^f \subseteq s_j' \subseteq \tilde{s_j}$.

[1] Keep in mind that $s_1,\ldots,s_n,\pi,\phi,\tilde{s_1},\ldots,\tilde{s_n}$, and α_i whenever $* \in s_i$, are fixed *before* the inductive proof of P.

1. Assume that $(s_1,\dots,s_n,\pi) \models \triangle_i T$, i.e. $[T] \subseteq s_i$. I show $L(\triangle_i T)$ in the following three cases:

   i) $* \notin s_i$ and s_i finite: let $s_i^f = s_i$, then a), b) and c) hold trivially.

   ii) $* \notin s_i$ and s_i infinite: by Lemma 6.5, since $[T]$ is a finite subset of s_i, there is a $s_i^f \in S_i \cap \wp^{fin}(s_i)$, giving a) and b) ($\tilde{s_i} = s_i$), such that $[T] \subseteq s_i^f$. If $s_i^f \subseteq s_i' \subseteq \tilde{s_i}$, then $[T] \subseteq s_i'$ and $(s_1',\dots,s_n',\pi) \models \triangle_i T$ giving c).

   iii) $* \in s_i$: by Lemma 6.7.1 (with $S = S_i$ and $s = s_i$), since $[T] \subseteq s_i^\phi$ and $[T] \subseteq (s_i \setminus \{*\})$, there is a $s_i^f \in S_i \cap \wp^{fin}(\tilde{s_i})$, giving a) and b) for i, such that $[T] \subseteq s_i^f$. If $s_i^f \subseteq s_i' \subseteq \tilde{s_i}$, then $[T] \subseteq s_i'$ and $(s_1',\dots,s_n',\pi) \models \triangle_i T$ giving c).

2. Assume that $(s_1,\dots,s_n,\pi) \models \neg \triangle_i T$, i.e. $[T] \not\subseteq s_i$. I show $L(\neg \triangle_i T)$. By (6.9) there exists a s_i^f such that $s_i^f \in S_i^f$ and $s_i^f \subseteq \tilde{s_i}$, giving a) and b). Let s_i' be such that $s_i^f \subseteq s_i' \subseteq \tilde{s_i}$. First, consider that $* \notin s_i$. Then $\tilde{s_i} = s_i$, and since $[T] \not\subseteq s_i$ $[T] \not\subseteq s_i'$. Second, consider that $* \in s_i$. Assume that $[T] \subseteq s_i'$. Then $[T] \subseteq \tilde{s_i}$ but, since $\triangle_i T \leq \phi$, $\alpha_i \notin [T]$ by definition of α_i, so $[T] \subseteq (s_i \setminus \{*\})$. But this is a contradiction, since $(s_i \setminus \{*\}) \subset s_i$ and $[T] \not\subseteq s_i$, so the assumption that $[T] \subseteq s_i'$ is impossible. Thus in either case, $[T] \not\subseteq s_i'$, and $(s_1',\dots,s_n',\pi) \models \neg \triangle_i T$ giving c).

$\psi = \triangledown_i T$: Assume that $\triangledown_i T \leq \phi$. In both the following cases, let s_j^f, $j \neq i$, be such that $s_j^f \in S_j^f$ and $s_j^f \subseteq \tilde{s_j}$ (giving a) and b) for $j \neq i$) – existence of such s_j^f is given by (6.9) – and let s_j' be arbitrary such that $s_j^f \subseteq s_j' \subseteq \tilde{s_j}$.

1. Assume that $(s_1,\dots,s_n,\pi) \models \triangledown_i T$, i.e. $s_i \subseteq [T]$. I show $L(\triangledown_i T)$. Simply choosing $s_i^f = s_i$ suffice: a) holds since $s_i \in S_i$ and s_i is finite since $s_i \subseteq [T]$. b) holds since $s_i \subseteq [T] \Rightarrow * \notin s_i \Rightarrow \tilde{s_i} = s_i = s_i^f \Rightarrow s_i^f \subseteq \tilde{s_i}$. Let s_i' be such that $s_i^f \subseteq s_i' \subseteq \tilde{s_i}$. Since $s_i^f = \tilde{s_i}$, $s_i' = \tilde{s_i} = s_i$. $s_i' \subseteq [T]$, and $(s_1',\dots,s_n',\pi) \models \triangledown_i T$.

2. Assume that $(s_1,\dots,s_n,\pi) \models \neg \triangledown_i T$, i.e. $s_i \not\subseteq [T]$. I show $L(\neg \triangledown_i T)$ in the following three cases:

   i) $* \notin s_i$ and s_i finite: let $s_i^f = s_i$, then a), b) and c) hold trivially.

   ii) $* \notin s_i$ and s_i infinite: $\tilde{s_i} = s_i$. By Lemma 6.6.1.b, since $[T] \in \wp^{fin}(OL)$, there is a s_i^f such that $s_i^f \in S_i^f$ and $s_i^f \subset s_i = \tilde{s_i}$ giving a) and b), and such that $s_i^f \not\subseteq [T]$. If s_i' is such that $s_i^f \subseteq s_i' \subseteq \tilde{s_i} = s_i$, then $s_i' \not\subseteq [T]$ and $(s_1',\dots,s_n',\pi) \models \neg \triangledown_i T$ giving c).

   iii) $* \in s_i$: Since $[T] \subseteq s_i^\phi$, by Lemma 6.7.2 there is a s_i^f such that $s_i^f \in S_i^f$ and $s_i^f \subset \tilde{s_i}$, giving a) and b), and such that $s_i^f \not\subseteq [T]$. If s_i' is such that $s_i^f \subseteq s_i' \subseteq \tilde{s_i}$, then $s_i' \not\subseteq [T]$ and $(s_1',\dots,s_n',\pi) \models \neg \triangledown_i T$ giving c).

$\psi = \neg\psi_1$: The induction hypothesis is $P(\psi_1)$. Assume that $\neg\psi_1 \leq \phi$; then also $\psi_1 \leq \phi$.

1. Assume that $(s_1, \ldots, s_n, \pi) \models \neg\psi_1$. Then, since $\psi_1 \leq \phi$, $L(\neg\psi_1) = L(\psi)$ holds by $P(\psi_1)$ 2).
2. Assume that $(s_1, \ldots, s_n, \pi) \models \neg\neg\psi_1$. Then, $(s_1, \ldots, s_n, \pi) \models \psi_1$ and since $\psi_1 \leq \phi$, $L(\psi_1)$ holds by $P(\psi_1)$ 1). By the definition of L, $L(\psi_1)$ holds iff $L(\neg\psi) = L(\neg\neg\psi_1)$ holds.

$\psi = \psi_1 \wedge \psi_2$**:** The induction hypotheses are $P(\psi_1)$ and $P(\psi_2)$. Assume that $\psi_1 \wedge \psi_2 \leq \phi$.

1. Assume that $(s_1, \ldots, s_n, \pi) \models \psi_1 \wedge \psi_2$. $\psi_1 \wedge \psi_2 \leq \phi$ implies that $\psi_1 \leq \phi$, and $(s_1, \ldots, s_n, \pi) \models \psi_1 \wedge \psi_2$ implies that $(s_1, \ldots, s_n, \pi) \models \psi_1$, and thus, by $P(\psi_1)$, $L(\psi_1)$ holds. That is, there exist, for each agent i, $s^f_{1,i} \in S^f_i$ such that $s^f_{1,i} \subseteq \tilde{s}_i$ and for all $s'_{1,i}$ such that $s^f_{1,i} \subseteq s'_{1,i} \subseteq \tilde{s}_i$, $(s'_{1,1}, \ldots, s'_{1,n}, \pi) \models \psi_1$. Similarly, by $P(\psi_2)$, $L(\psi_2)$ holds; there exist, for each agent i, $s^f_{2,i} \in S^f_i$ such that $s^f_{2,i} \subseteq \tilde{s}_i$ and for all $s'_{2,i}$ such that $s^f_{2,i} \subseteq s'_{2,i} \subseteq \tilde{s}_i$, $(s'_{2,1}, \ldots, s'_{2,n}, \pi) \models \psi_2$. I show $L(\psi_1 \wedge \psi_2)$. Since $S_i \cap \wp^{fin}(\tilde{s}_i)$ is directed (by Def. 6.8.1.a when $* \notin s_i$ and by Lemma 6.7.3 when $* \in s_i$ (recall that s_i is finite when $* \in s_i$)) and $s^f_{1,i}, s^f_{2,i} \in S_i \cap \wp^{fin}(\tilde{s}_i)$, there exists, for each i a $s^f_i \in S_i \cap \wp^{fin}(\tilde{s}_i)$ such that $s^f_{1,i}, s^f_{2,i} \subseteq s^f_i$. a) holds since $s^f_i \in S_i$ is finite, and b) holds since $s^f_i \in \wp^{fin}(\tilde{s}_i)$. Let, for each i, s'_i be such that $s^f_i \subseteq s'_i \subseteq \tilde{s}_i$. Because $s^f_{1,i} \subseteq s^f_i \subseteq s'_i \subseteq \tilde{s}_i$, $s^f_{1,i} \subseteq s'_i \subseteq \tilde{s}_i$ and, by $L(\psi_1)$, $(s'_1, \ldots, s'_n, \pi) \models \psi_1$. Similarly, because $s^f_{2,i} \subseteq s^f_i \subseteq s'_i \subseteq \tilde{s}_i$, $s^f_{2,i} \subseteq s'_i \subseteq \tilde{s}_i$ and, by $L(\psi_2)$, $(s'_1, \ldots, s'_n, \pi) \models \psi_2$. Thus, $(s'_1, \ldots, s'_n, \pi) \models \psi_1 \wedge \psi_2$, and c) holds.
2. Assume that $(s_1, \ldots, s_n, \pi) \models \neg(\psi_1 \wedge \psi_2)$; $(s_1, \ldots, s_n, \pi) \models \neg\psi_1 \vee \neg\psi_2$; $(s_1, \ldots, s_n, \pi) \models \neg\psi_1$ or $(s_1, \ldots, s_n, \pi) \models \neg\psi_2$. Assume the first case (the proof in the second case is symmetrical). $\psi_1 \wedge \psi_2 \leq \phi$ implies that $\psi_1 \leq \phi$ and since $(s_1, \ldots, s_n, \pi) \models \neg\psi_1$, $L(\neg\psi_1)$ holds by $P(\psi_1)$. That is, there exist $s^f_i \in S^f_i$ such that $s^f_i \subseteq \tilde{s}_i$ and for all s'_i such that $s^f_i \subseteq s'_i \subseteq \tilde{s}_i$, $(s'_1, \ldots, s'_n, \pi) \models \neg\psi_1$. But then we also have that $(s'_1, \ldots, s'_n, \pi) \models \neg(\psi_1 \wedge \psi_2)$ (i.e. s^f_i, the witness in $L(\neg\psi_1)$, is also a witness in $L(\neg(\psi_1 \wedge \psi_2))$). ■

Recall that a set Φ of epistemic axioms induces sets of legal epistemic states S^Φ_i (Defs. 6.3 and 6.4).

Theorem 6.1 If Φ is a set of epistemic axioms such that $S^\Phi_1, \ldots, S^\Phi_n$ are finitary sets of epistemic states, then Φ is finitary. □

PROOF Since Φ are epistemic axioms, $\mathcal{M}^\Phi_{Ax} = \{(s^\Phi_1, \ldots, s^\Phi_n, \pi) \in \mathcal{M} : s^\Phi_i \in S^\Phi_i\}$. Since all S^Φ_i are finitary, by Lemma 6.8 $\mathcal{M}^\Phi_{Ax}$ is a finitary set of GKSSs. Since $\mathcal{M}^\Phi_{Ax} = mod(\Phi)$ (Lemma 6.2), Φ is finitary by Lemma 6.4. ■

Theorem 6.1 shows that the conditions in Def. 6.8 on the set of legal epistemic states induced by epistemic axioms are sufficient to conclude that the axioms are finitary.

6.6.1 Stronger Conditions

I have (Def. 6.8) given sufficient conditions on the sets of epistemic states induced by epistemic axioms, for the axioms to be finitary. In this section, I present several alternative sufficient conditions which are stronger (i.e. imply Def. 6.8).

Lemma 6.9 A set of epistemic states $S \subseteq \mathcal{S}$ is finitary if

1. For every infinite $s \in S$:
   (a) $S|_s^f$ is directed
   (b) $S|_s^f$ is a cover of s
2. $\forall_{s \cup \{*\} \in S} \forall_{s' \in \wp^{fin}(OL)} \exists_{\alpha \notin s'}$:
   (a) $S|_{s\cup\{\alpha\}}^f$ is directed
   (b) $S|_{s\cup\{\alpha\}}^f$ is a cover of $s \cup \{\alpha\}$ □

PROOF It must be shown that 2 in this Lemma implies 2 in Def. 6.8. Assume that 2 holds, and let $s \cup \{*\} \in S$, $s' \in \wp^{fin}(OL)$ and α as described in 2 above.

Def. 6.8.2a) $s' \cap s$ is finite, say $s' \cap s = \{\beta_1, \ldots, \beta_k\}$. By 2b), $s \cup \{\alpha\} \subseteq \bigcup(S \cap \wp^{fin}(s \cup \{\alpha\}))$. Since $s' \cap s \subseteq s \cup \{\alpha\}$, $\{\beta_1, \ldots, \beta_k\} \subseteq \bigcup(S \cap \wp^{fin}(s \cup \{\alpha\}))$. That is, there exist $s_j \in S \cap \wp^{fin}(s \cup \{\alpha\})$ for $1 \leq j \leq k$ such that $\beta_j \in s_j$. By 2a), there is a $s^f \in S \cap \wp^{fin}(s \cup \{\alpha\})$ such that $\bigcup_{j=1}^k s_j \subseteq s^f$. Since $\{\beta_1, \ldots, \beta_k\} \subseteq \bigcup_{j=1}^k s_j$, $s' \cap s \subseteq s^f$ and Def. 6.8.2a) holds.

Def. 6.8.2b) Assume that Def. 6.8.2b) does *not* hold, i.e. that $\forall_{s^f \in S \cap \wp(s \cup \{\alpha\})} s^f \subseteq s'$. Then, $\bigcup(S \cap \wp^{fin}(s \cup \{\alpha\})) \subseteq s'$. Since $\alpha \notin s'$, $s \cup \{\alpha\} \not\subseteq s'$ and $s \cup \{\alpha\} \not\subseteq \bigcup(S \cap \wp^{fin}(s \cup \{\alpha\}))$, contradicting 2b) of this Lemma. Thus, Def. 6.8.2b) must hold.

Def. 6.8.2c) 2a) of this Lemma. ■

Corollary 6.1 Let $S \subseteq \mathcal{S}$ be a set of epistemic states. The following are conditions on S together with implications on the conditions in Lemma 6.9:

1. If $S|_s^f$ is directed for all $s \subseteq OL$:
   - Lemma 6.9.1a) holds
   - Lemma 6.9.2a) holds (for *all* α)
2. If $S|_s^f$ is a cover of s for all $s \subseteq OL$:
   - Lemma 6.9.1b) holds
   - Lemma 6.9.2b) holds (for *all* α)
3. If $\{\alpha\} \in S$ for every $\alpha \in OL$:

- Lemma 6.9.1b) holds
- Lemma 6.9.2b) holds (for *all* α)

4. If $\forall_{s\cup\{*\}\in S}\forall_{s'\in\wp^{fin}(OL)}\exists_{\alpha\notin s'} s\cup\{\alpha\}\in S$:
   - Lemma 6.9.2 holds □

PROOF

1. Follows immediately (in Lemma 6.9.2a: for *any* $\alpha \in OL$)
2. Follows immediately (in Lemma 6.9.2b: for *any* $\alpha \in OL$)
3. **Lemma 6.9.1b)** : Let $s \in S$ be infinite. If $\{\beta\} \in s$, then $\beta \in \wp^{fin}(s)$ and $\{\beta\} \in S$, thus $s \subseteq \bigcup(S \cap \wp^{fin}(s))$.

   Lemma 6.9.2b) : Let $s\cup\{*\} \in S$, $s' \in \wp^{fin}(OL)$ and $\alpha \in OL$ be arbitrary such that $\alpha \notin s'$. If $\beta \in s\cup\{\alpha\}$, then $\{\beta\} \in \wp^{fin}(s\cup\{\alpha\})$ and $\{\beta\} \in S$, thus $s\cup\{\alpha\} \subseteq \bigcup(S\cap\wp^{fin}(s\cup\{\alpha\}))$.
4. Let $s\cup\{*\} \in S$, $s' \in \wp^{fin}(OL)$ and $\alpha \in OL$ be such that $\alpha \notin s'$ and $s\cup\{\alpha\} \in S$.

   Let $s^f = s\cup\{\alpha\}$. Clearly, $s^f \in S\cap\wp^{fin}(s\cup\{\alpha\})$, and if $s_1, s_2 \in S\cap\wp^{fin}(s\cup\{\alpha\})$ then $s_1, s_2 \subseteq s^f$ and Lemma 6.9.2a) holds.

   Lemma 6.9.2b) holds trivially, since $s\cup\{\alpha\} \subseteq \bigcup(S\cap\wp^{fin}(s\cup\{\alpha\}))$. ■

The following corollary collects some of the possible combinations of the conditions.

Corollary 6.2 A set of epistemic states $S \subseteq \mathcal{S}$ is finitary if either one of the following three conditions hold:

1. For every $s \subseteq OL$:
   (a) $S|_s^f$ is directed
   (b) $S|_s^f$ is a cover of s
2. (a) $S|_s^f$ is directed for every $s \subseteq OL$
   (b) $\{\alpha\} \in S$ for every $\alpha \in OL$
3. (a) $S|_s^f$ is directed for every infinite $s \in S$
   (b) $\{\alpha\} \in S$ for every $\alpha \in OL$
   (c) $\forall_{s\cup\{*\}\in S}\forall_{s'\in\wp^{fin}(OL)}\exists_{\alpha\notin s'} s\cup\{\alpha\}\in S$ □

PROOF

1. Follows from Lemma 6.9 and Corollary 6.1.1 and 2.
2. Follows from Lemma 6.9 and Corollary 6.1.1 and 3.
3. Follows from Lemma 6.9 and Corollary 6.1.3 and 4. ■

Finitaryness of the Empty Theory

In Chapter 5 the result that the empty set is finitary, and hence that the calculus *EC* is weakly complete, was stated (Lemma 5.2) without proof. The result is now proved by using the finitaryness conditions developed in this section.

Lemma 6.10 (Lemma 5.2) The empty theory is finitary. □

PROOF I use Corollary 6.2.1 to show that $\mathcal{S}$, the set of all epistemic states, is finitary. Let $s \subseteq OL$. $\mathcal{S}|_s^f = \mathcal{S} \cap \wp^{fin}(s) = \wp^{fin}(s)$. $\wp^{fin}(s)$ is directed, because for every finite subset $B \subset \wp^{fin}(s)$, $\cup_{s' \in B} s' \in \wp^{fin}(s)$. $\wp^{fin}(s)$ is a cover of s, because $s \subseteq \bigcup \wp^{fin}(s)$.

Thus, by Corollary 6.2.1, $\mathcal{S}$ is finitary. By definition (Def. 6.2) the empty set is a set of epistemic axioms, and $S_i^{\emptyset} = \mathcal{S}$ for each i (Def. 6.4). By Theorem 6.1, $\emptyset$ is finitary. ■

6.7 Examples

In this section, I look at some examples of extensions of the basic framework. For selected axioms, I construct the model classes as described in Section 6.3.1 and investigate finitaryness properties as discussed in Section 6.6.

Written in SSEL notation, the commonly used axioms of modal epistemic logic discussed in Section 2.1 are:

$\triangle_i \{(\alpha \to \beta)\} \to (\triangle_i\{\alpha\} \to \triangle_i\{\beta\})$	Distribution	$\mathbf{K_i}$
$\triangle_i \{\alpha\} \to \neg \triangle_i \{\neg\alpha\}$	Consistency	$\mathbf{D_i}$
$\triangle_i \{\alpha\} \to \alpha$	Knowledge	$\mathbf{T_i}$
$\triangle_i \{\alpha\} \to \triangle_i\{\triangle_i\{\alpha\}\}$	Positive Introspection	$\mathbf{4_i}$
$\neg \triangle_i \{\alpha\} \to \triangle_i\{\neg \triangle_i \{\alpha\}\}$	Negative Introspection	$\mathbf{5_i}$

In order to construct the model classes for these axioms, except for $\mathbf{T_i}$ which is not an epistemic axiom, we can interpret the axioms as schemata over both the agent index i and AL formulae α, β. These axiom schemata must be viewed as sets of axioms:

$$\begin{aligned}
\mathbf{K} &= \{K_i(\alpha,\beta) : 1 \le i \le n, \alpha,\beta \in AL\} \\
&\quad \text{where } K_i(\alpha,\beta) = \triangle_i\{(\alpha \to \beta)\} \to (\triangle_i\{\alpha\} \to \triangle_i\{\beta\}) \\
\mathbf{D} &= \{D_i(\alpha) : 1 \le i \le n, \alpha \in AL\} \\
&\quad \text{where } D_i(\alpha) = \triangle_i\{\alpha\} \to \neg \triangle_i \{\neg\alpha\} \\
\mathbf{4} &= \{4_i(\alpha) : 1 \le i \le n, \alpha \in AL\} \\
&\quad \text{where } 4_i(\alpha) = \triangle_i\{\alpha\} \to \triangle_i\{\triangle_i\{\alpha\}\} \\
\mathbf{5} &= \{5_i(\alpha) : 1 \le i \le n, \alpha \in AL\} \\
&\quad \text{where } 5_i(\alpha) = \neg \triangle_i \{\alpha\} \to \triangle_i\{\neg \triangle_i \{\alpha\}\}
\end{aligned}$$

The model classes are defined by (see Section 6.3.1)

$$\mathcal{M}_{Ax}^{\Phi} = \{(s_1^{\Phi}, \ldots, s_n^{\Phi}, \pi) : s_i^{\Phi} \in S_i^{\Phi}\}$$

for $\Phi \in \{\mathbf{K}, \mathbf{D}, \mathbf{4}, \mathbf{5}\}$, where:

$$\begin{array}{ll}
S_i^{K_i(\alpha,\beta)} & = \mathcal{S} \setminus \{X \in \mathcal{S} : [\alpha \to \beta] \in X\} \cup (\mathcal{S} \setminus \{X \in \mathcal{S} : [\alpha] \in X\} \\
& \cup \{X \in \mathcal{S} : [\beta] \in X\}) \\
& = \mathcal{S} \setminus \{X \in \mathcal{S} : ([\alpha] \to [\beta]), [\alpha] \in X, [\beta] \notin X\} \\
S_i^{D_i(\alpha)} & = \mathcal{S} \setminus \{X \in \mathcal{S} : [\alpha] \in X\} \cup \mathcal{S} \setminus \{X \in \mathcal{S} : \neg[\alpha] \in X\} \\
& = \mathcal{S} \setminus \{X \in \mathcal{S} : [\alpha], \neg[\alpha] \in X\} \\
S_i^{4_i(\alpha)} & = \mathcal{S} \setminus \{X \in \mathcal{S} : [\alpha] \in X\} \cup \{X \in \mathcal{S} : \triangle_i\{[\alpha]\} \in X\} \\
& = \mathcal{S} \setminus \{X \in \mathcal{S} : [\alpha] \in X, \triangle_i\{[\alpha]\} \notin X\} \\
S_i^{5_i(\alpha)} & = \mathcal{S} \setminus (\mathcal{S} \setminus \{X \in \mathcal{S} : [\alpha] \in X\}) \cup \{X \in \mathcal{S} : \neg \triangle_i \{[\alpha]\} \in X\} \\
& = \mathcal{S} \setminus \{X \in \mathcal{S} : [\alpha] \notin X, \neg \triangle_i \{[\alpha]\} \notin X\} \\
S_i^{\mathbf{K}} & = \bigcap_{K_i(\alpha,\beta) \in \mathbf{K}} S_i^{K_i(\alpha,\beta)} = \\
& \mathcal{S} \setminus \{X \in \mathcal{S} : \exists \alpha, \beta \in AL(([\alpha] \to [\beta]), [\alpha] \in X, [\beta] \notin X)\} \\
S_i^{\mathbf{D}} & = \bigcap_{D_i(\alpha) \in \mathbf{D}} S_i^{D_i(\alpha)} = \mathcal{S} \setminus \{X \in \mathcal{S} : \exists \alpha \in AL([\alpha], \neg[\alpha] \in X)\} \\
S_i^{\mathbf{4}} & = \bigcap_{4_i(\alpha) \in \mathbf{4}} S_i^{4_i(\alpha)} = \mathcal{S} \setminus \{X \in \mathcal{S} : \exists \alpha \in AL([\alpha] \in X, \triangle_i\{[\alpha]\} \notin X)\} \\
S_i^{\mathbf{5}} & = \bigcap_{5_i(\alpha) \in \mathbf{5}} S_i^{5_i(\alpha)} = \mathcal{S} \setminus \{X \in \mathcal{S} : \exists \alpha \in AL([\alpha] \notin X, \neg \triangle_i \{[\alpha]\} \notin X)\}
\end{array}$$

I next investigate the finitaryness of these axioms.

Lemma 6.11

1. **K** is a finitary theory
2. **D** is a finitary theory
3. **4** is not a finitary theory
4. **5** is not a finitary theory □

PROOF 1. I show that $S_i^{\mathbf{K}}$ is finitary for arbitrary i, by using Corollary 6.2.3. **K** is then finitary by Theorem 6.1.

Corollary 6.2.3.(a): I show that $S_i^{\mathbf{K}}|_s^f$ is directed for infinite $s \in S_i^{\mathbf{K}}$. Let $s', s'' \in S_i^{\mathbf{K}} \cap \wp^{fin}(s)$, and let:

$$\begin{array}{l}
s_0 = s' \cup s'' \\
s_j = s_{j-1} \cup \{[\beta] : [\alpha \to \beta], [\alpha] \in s_{j-1}\} \quad 0 < j \\
s^f = \bigcup_j s_j
\end{array}$$

If there are formulae $\alpha \to \beta, \alpha$ such that $[\alpha] \to [\beta], [\alpha] \in s^f$, then there are s_l and s_m such that $[\alpha] \to [\beta] \in s_l$ and $[\alpha] \in s_m$. Then, $[\beta] \in s_{max(l,m)+1}$ and $[\beta] \in s^f$. Thus, $s^f \in S_i^{\mathbf{K}}$.

I show that $s_j \in \wp^{fin}(s)$ for every j by induction over j. $s_0 \in \wp^{fin}(s)$, since $s', s'' \in \wp^{fin}(s)$. Assume that $s_{j-1} \in \wp^{fin}(s)$, $j > 0$. If $s_j = s_{j-1}$ then $s_j \in \wp^{fin}(s)$. Otherwise, let $[\beta] \in s_j$ such that $[\beta] \notin s_{j-1}$ and $[\alpha] \to [\beta], [\alpha] \in s_{j-1}$ for some α. Since $s_{j-1} \subseteq s$, $[\alpha] \to [\beta], [\alpha] \in s$, and since $s \in S_i^{\mathbf{K}}$, $[\beta] \in s$. Since $[\beta] \in s$ for every $[\beta] \in s_j \setminus s_{j-1}$ and $s_{j-1} \subseteq s$, $s_j \subseteq s$. Furthermore, $s_j \setminus s_{j-1}$ is finite since s_j is finite, and therefore $s_j \in \wp^{fin}(s)$. Thus, $s_j \in \wp^{fin}(s)$ for every j, and $s^f \subseteq s$.

It is easy to see that $[\beta] \in s_j \setminus s_{j-1}$ if and only if there is a $[\alpha_0] \rightarrow ([\alpha_1] \rightarrow \ldots \rightarrow ([\alpha_p] \rightarrow [\beta]) \cdots) \in s_0$ for some p. Since s_0 is finite, there is a $k > 0$ such that $s_j \setminus s_{j-1} = \emptyset$ for all $j \geq k$. s^f is the union of a finite number of finite sets, so s^f is finite. Thus, $s^f \in \wp^{fin}(s)$.

Since $s', s'' \subseteq s^f$ and $s^f \in S_i^{\mathbf{K}} \cap \wp^{fin}(s)$, $S_i^{\mathbf{K}}|_s^f$ is directed.

Corollary 6.2.3.(b): Clearly, $\{\alpha\} \in S_i^{\mathbf{K}}$ for every $\alpha \in OL$.

Corollary 6.2.3.(c): Let $s \cup \{*\} \in S_i^{\mathbf{K}}$ and $s' \in \wp^{fin}(OL)$. Let $\alpha \in OL$ be such that:

- $\alpha \rightarrow \beta \notin s$ for any $\beta \in OL$
- $\alpha \notin s'$
- The main connective in α is not implication

It is easy to see that there exist infinitely many α satisfying these three conditions; there are infinitely many $\alpha \in OL$ without implication as main connective, and both s and s' are finite.

I show that $s \cup \{\alpha\} \in S_i^{\mathbf{K}}$. Let $\alpha' \rightarrow \beta, \alpha' \in s \cup \{\alpha\}$. Since α does not have implication as main connective, $\alpha' \rightarrow \beta \in s$. Since α is such that $\alpha \rightarrow \beta' \notin s$ for any β', $\alpha' \neq \alpha$ and $\alpha' \in s$. Then, since $s \cup \{*\} \in S_i^{\mathbf{K}}$, $\beta \in s$. Thus, $s \cup \{\alpha\} \in S_i^{\mathbf{K}}$.

2. I show that $S_i^{\mathbf{D}}$ is finitary for arbitrary i, by using Corollary 6.2.3. **D** is then finitary by Theorem 6.1.

Corollary 6.2.3.(a): I show that $S_i^{\mathbf{D}}|_s^f$ is directed for infinite $s \in S_i^{\mathbf{D}}$. Let $s', s'' \in S_i^{\mathbf{D}} \cap \wp^{fin}(s)$, and let $s^f = s' \cup s''$. If there is a β such that $[\beta], \neg[\beta] \in s^f$, then $[\beta], \neg[\beta] \in s$, which is impossible since $s \in S_i^{\mathbf{D}}$. Thus, $s^f \in S_i^{\mathbf{D}}$. Since $s', s'' \in \wp^{fin}(s)$, $s^f \in \wp^{fin}(s)$.

Thus, there is a $s^f \in S_i^{\mathbf{D}} \cap \wp^{fin}(s)$ such that $s', s'' \subseteq s^f$.

Corollary 6.2.3.(b): Clearly, $\{\alpha\} \in S_i^{\mathbf{D}}$ for every $\alpha \in OL$.

Corollary 6.2.3.(c): Let $s \cup \{*\} \in S_i^{\mathbf{D}}$ and $s' \in \wp^{fin}(OL)$. Let $\alpha \in OL$ be such that:

- $\neg\alpha \notin s$
- $\alpha \notin s'$
- α does not start with negation

It is easy to see that there exist infinitely many α satisfying these three conditions; there are infinitely many $\alpha \in OL$ without negation as main connective, and both s and s' are finite.

I show that $s \cup \{\alpha\} \in S_i^{\mathbf{D}}$. Assume the opposite, i.e. that $\beta, \neg\beta \in s \cup \{\alpha\}$. Since α does not start with negation, $\neg\beta \in s$. Since $\neg\alpha \notin s$, $\beta \neq \alpha$ and $\beta \in s$. But since $s \in S_i^{\mathbf{D}}$ it is a contradiction that $\beta, \neg\beta \in s$. Thus, $s \cup \{\alpha\} \in S_i^{\mathbf{D}}$.

3. Let $1 \leq i \leq n$, and let $M = (s_1, \ldots, s_n, \pi) \in \mathcal{M}_{fin}$ such that $M \models_f \mathbf{4}$. Clearly, s_i must be the empty set – otherwise it would not be finite. Thus, $\mathbf{4} \models_f \triangledown_i\emptyset$. $\mathbf{4}$ does, however, have *infinite* models, so $\mathbf{4} \not\models \triangledown_i\emptyset$. Lemma 5.3 gives that $\mathbf{4}$ is not finitary.

4. It is easy to see that **5** is not satisfiable in $\mathcal{M}_{\mathit{fin}}$ (i.e. that a model for **5** must be infinite). By Theorem 5.1 and Lemma 5.1, **5** is not finitary. ■

Since we only have semantic finitaryness conditions for epistemic axioms, we cannot use them to decide the finitaryness of **T**, which is not a set of epistemic axioms.

Part III

A Dynamic Logic of Finite Syntactic Epistemic States

Chapter 7

Language and Semantics

7.1 Introduction

In Part II a logic of syntactic epistemic states at a given point in time was presented. Agents were represented as points in the lattice of sets of object formulae, and the logical language could express facts about their knowledge. Contrary to other epistemic formalisms, it does not follow from only the fact that agents know certain formulae that they also must know — now or in the future — certain other formulae. In Part III, the language will now be extended to be able to express that agents can know *rules* in addition to formulae. In addition, temporal operators are added to express facts about possible futures. Together, the facts that an agent knows certain formulae and certain rules may imply that he *can* come to know other formulae in the future. Or, the facts that he does not know certain formulae now and *only knows* certain rules may imply that he *cannot* know some other formulae in the future.

Semantically, the formalism is extended to consider more than one given point in the lattice for each agent by allowing the agents to move in the lattice. The language which constitutes the agents' epistemic states (*OL*) (and thus its syntactic representation *AL*) is not changed from Part II, and the epistemic states are required to be finite. The following additional semantic assumptions are made. Time is discrete. An agent's epistemic state can change between points in time. In addition to a syntactical storage of formulae, each agent has a *mechanism* for transforming sets of formulae — e.g. making deductions, belief revision, forgetting, etc. In addition to *reasoning*, the mechanism can also do *communication*; e.g. send (syntactic) formulae to other agents. Communication is seen as a generalization of reasoning. Between two points in time, *all* agents use their mechanisms *simultaneously*[1].

Since agents can know several rules and their mechanisms are not required to be deterministic, the future is not deterministic. Thus a logic of future epistemic states should be a *branching time* logic in which e.g. what *may* happen and what *must* happen can be expressed. Since agents can communicate, they can cooperate. For example, if an agent knows modus ponens and the formula $p \to q$ and another agent knows the formula p, the two agents can cooperate

[1]Asynchronous systems can be modelled with the help of mechanisms which do not always change the current epistemic state.

to make the first agent know q. This type of branching time cooperation logic is exactly what ATL (see Section 2.3) is. The logic developed in the current chapter is an ATL logic, in the sense that the language can be interpreted as an ATL language and mechanisms as concurrent game structures (integration of epistemic logic and ATL is not a new idea; see Section 9.4).

In the next section, the two new parts of the logical language are introduced: rules for reasoning and communication and ATL type temporal connectives for expressing facts about (branching time) futures. The interpretation of rules as a relation between epistemic states is defined in Section 7.3. In Section 7.4 the properties of the concurrent game structures we are interested in are discussed, followed by a model of agents' mechanism and a definition of the concurrent game structure induced by a mechanism in Section 7.5. In Section 7.6 satisfiability in a mechanism is defined as satisfiability in the induced concurrent game structure and the relation between the notions of having a mechanism and knowing a set of rules is discussed in detail. The resulting logic is called *Dynamic Syntactic Epistemic Logic (*DSEL*)*. An example study is presented in Chapter 8.

As already mentioned, DSEL is an extension of SSEL (Static Syntactic Epistemic Logic) from Part II both syntactically and semantically: the logical language is an extension of EL and each point in time is a structure for EL. The extended formalism is not developed to the same extent, however; a calculus is left for future work.

7.2 Language

In Part II I defined a logical meta-language and an agent language. I now extend the meta language for expressing dynamical properties of reasoning and communication, while keeping the same agent language.

The syntax of *rules* is defined, and the meta language extended with

1. Epistemic operators for rules
2. Temporal connectives

Both the epistemic rule operators and temporal connectives can express properties of an agent's *mechanism* which allows him to change his own and other agents' epistemic states. The new epistemic operators allow us to express that an agent *knows* a rule (e.g. *modus ponens*). The new temporal operators allow us to express statements about the future, e.g. that an agent may get to know a certain formula.

7.2.1 Rules

Rules are defined over two new sets of *variables*: $V_F = \{a, b, c, \ldots\}$ and $V_T = \{t, u, v, \ldots\}$, used as placeholders for formulae and terms respectively. A rule consists of an *antecedent* and a *consequent*; both representing sets of formulae — possibly containing variables. Examples of rules are:

$$R_1 = \frac{t \sqcup \{a \to b, a\}}{t \sqcup \{b\}} \qquad R_2 = \frac{t \sqcup \{p\}}{t \sqcup \{\triangle_j\{p\}\}}$$

The intended meaning of "knowing a rule" is that if the current epistemic state matches the antecedent, then it can be changed to an epistemic state matching the consequent.

Rules are used in the meta-language by introducing new epistemic operators for rules.

$$\overset{\rightsquigarrow}{\triangle}_i\{R_1\}$$

denotes the fact that agent i knows at least the rule R_1 (but it may know more rules), and

$$\overset{\rightsquigarrow}{\triangledown}_i\{R_1, R_2, R_3\}$$

denotes the fact that at most rules R_1, R_2 and R_3 are known by i (but it may be the case that i does not know all of them).

Even though $\triangle_i$ and $\overset{\rightsquigarrow}{\triangle}_i$ ($\triangledown_i$ and $\overset{\rightsquigarrow}{\triangledown}_i$) are similar in appearance and have similar meanings, *rules are not formulae* — they have a different ontological status. Hence, it is not possible to write e.g. $\triangle_i\{p, R_1\}$ to denote the fact that agent i knows both the proposition p and the rule R_1 — this fact should be written $\triangle_i\{p\} \wedge \overset{\rightsquigarrow}{\triangle}_i\{R_1\}$. Furthermore, unlike formulae rules are not closed under propositional connectives; $\overset{\rightsquigarrow}{\triangle}_i\{R_1 \wedge R_2\}$ is not a well formed formula. Rules *only* appear as arguments to the operators $\overset{\rightsquigarrow}{\triangle}_i, \overset{\rightsquigarrow}{\triangledown}_i$, and they cannot be nested – the agent language is not extended with rule operators. For example, the formula

$$\triangle_i\{\overset{\rightsquigarrow}{\triangle}_j\{R_1\}\}$$

is not well formed.

Reasoning and Communication

We want to model agents' abilities to both reason and communicate. We also want to express these abilities in the logical language. The ability to reason is expressed by the epistemic rule operators; they are used to express the fact that an agent can go from one epistemic state (matching the antecedent) to another (matching the consequent). The ability to communicate can be expressed in a very similar manner: agent i can in a certain epistemic state perform a communication action resulting in a change in the epistemic state of agent j. Thus, communication can be seen as a generalization of reasoning.

To be able to express communication, the $\overset{\rightsquigarrow}{\triangle}$-operator just introduced is generalized to taking *two* subscripts:

$$\overset{\rightsquigarrow}{\triangle}_{ij}\{R_1\}$$

means that agent i knows at least the rule R_1 for communication to agent j.

$$\overset{\rightsquigarrow}{\triangledown}_{ij}\{R_1\}$$

means that agent i knows at most the rule R_1 for communication to agent j. Note that $\overset{\rightsquigarrow}{\triangledown}_{ij}\{R_1\}$ does not say anything about agent i's ability to communicate with agent k when $k \neq j$. The exact meaning of these rule operators will become clear in the following section.

The first version, denoting reasoning, is defined as a short-hand notation for the second; $\overset{\rightsquigarrow}{\triangle}_i \equiv \overset{\rightsquigarrow}{\triangle}_{ii}$. Similarly for $\overset{\rightsquigarrow}{\bigtriangledown}_i$.

7.2.2 Temporal Connectives

Three ATL-type temporal connectives (see Section 2.3) is introduced into the logical language, illustrated by the following examples. Let ϕ, ψ be formulae, and A a set of agents. The following are all well formed formulae.

$$\begin{gathered}\langle\langle A\rangle\rangle \bigcirc \phi \\ \langle\langle A\rangle\rangle \Box \phi \\ \langle\langle A\rangle\rangle \phi \mathcal{U} \psi\end{gathered}$$

Full nesting and propositional combinations of temporal formulas are allowed in the meta language but not in the agent language – the agent language is not extended with temporal connectives. The intended semantics is as in ATL: $\langle\langle A\rangle\rangle$ mens that "agents A can together make the following true", $\bigcirc$ means next, $\Box$ means globally, $\mathcal{U}$ means until. Derived connective:

$$\langle\langle A\rangle\rangle \mathcal{F} \psi \equiv \langle\langle A\rangle\rangle \mathbf{true} \mathcal{U} \psi$$

(the semantics of $\mathcal{U}$ requires ψ to eventually be true).

7.2.3 Example

The following is an example of a well formed formula, where $p, q \in \Theta$, $t \in V_T$ and $a, b \in V_F$:

$$\begin{aligned}&(\triangle_1\{p\} \wedge \triangle_2\{p \rightarrow q\} \\ &\quad \wedge \langle\langle\emptyset\rangle\rangle\Box \left(\overset{\rightsquigarrow}{\triangle}_{12}\{\frac{t \sqcup \{a\}}{\{a\}}\} \wedge \overset{\rightsquigarrow}{\triangle}_{22}\{\frac{t \sqcup \{a, a \rightarrow b\}}{t \sqcup \{a, a \rightarrow b, b\}}\} \wedge \overset{\rightsquigarrow}{\triangle}_{22}\{\frac{t}{t}\}\right)) \\ &\qquad\qquad\qquad\qquad \rightarrow \langle\langle\{1,2\}\rangle\rangle \mathcal{F} \triangle_2 \{q\}\end{aligned}$$

The intended meaning of this formula is that if agent 2 knows the rule modus ponens and the formula $p \rightarrow q$, and agent 1 knows the formula p and a rule for communicating any formula to 2, then agents 1 and 2 can cooperate to make 2 know the formula q in the future.

The use of the operator $\langle\langle\emptyset\rangle\rangle\Box$ was mentioned in Section 2.3.2 (p. 21), and it means that the following formula will always be true. Here, the consequence is that the rules will be remembered in the future. The use of the rule $\frac{t}{t}$ for agent 2 has the consequence that he can go to a new state without doing any inferences.

7.2.4 Formal Definitions

Here, the logical language *TEL* (Temporal Epistemic Language) is defined. To this end, several intermediate languages are defined. Formally, these languages are only used to define (and prove properties of) *TEL* and are not used as logical languages alone. The following convention is used for naming languages: a "*V*" subscript is used to denote a language extended with variables (representing formulae which are consequents and antecedents of rules), a “*T*” subscript is used to denote the term language (representing sets of formulae) induced by a language. Table 7.1 shows a summary of the formal language definitions which follow, and an example of the role of each of these intermediate languages in a *TEL* formula is shown in Fig. 7.1. Fig 7.2 further illustrates the relationships between the languages.

Language	**Description**
TEL	(Temporal) Epistemic Language. The logical (meta) language.
AL	The agent language, representing formulae agents can know.
AL_T	Terms representing sets of AL formulae.
AL_V	AL extended with formula-variables, used in antecedents and consequents of rules.
AL_{VT}	Terms representing sets of AL_V formulae.
TRL	(Temporal) Rule Language. A rule consists of two AL_{VT} terms: the consequent and the antecedent.
TRL_T	Terms representing sets of TRL rules.

Table 7.1: Overview and description of the formal language definitions involved in the definition of the main logical language *TEL*.

$$\underbrace{\overset{\rightsquigarrow}{\triangle}_i \underbrace{\{\underbrace{\overbrace{\frac{\{a, \overbrace{a \to b}^{AL_V}\}}{b}}^{AL_{VT}}}_{TRL}\}}_{TRL_T}\}}_{TEL} \wedge \underbrace{\triangle_i \overbrace{\{p, \overbrace{p \to q}^{AL}\}}^{AL_T}}_{TEL}$$

Figure 7.1: Illustration of the syntax of a *TEL* formula.

$$\begin{tikzcd}
 & & & & & V_T \arrow[d] & V_F \arrow[d] \\
AL \arrow[r, shift left] & AL_T \arrow[l, shift left] \arrow[r] & TEL & TRL_T \arrow[l] & TRL \arrow[l] & AL_{VT} \arrow[l] \arrow[r, shift left] & AL_V \arrow[l, shift left] \\
 & & \Theta \arrow[ull] \arrow[u] \arrow[urrrr] & & & &
\end{tikzcd}$$

Figure 7.2: Relationships between the intermediate languages used to define *TEL*. An arrow $A \longrightarrow B$ indicates that the language A is used directly in the definition of the language B.

Def. 3.2 provides a term language $TL(L)$ representing finite sets of a set of formulae L. In order to incorporate term-variables into terms, that definition is slightly extended[2] in the following one.

Definition 7.1 ($TL(L, S)$) Given a set of formulae L and a set of (atomic) terms S, the *term language* $TL(L, S)$ is the least set such that

- $S \subseteq TL(L, S)$
- If $\alpha_1, \dots, \alpha_k \in L$ then $\underline{\{}\alpha_1\underline{,}\dots\underline{,}\alpha_k\underline{\}} \in TL(L, S)$
- If $T, U \in TL(L, S)$ then $\left.\begin{array}{l}(T \sqcup U)\\(T \sqcap U)\end{array}\right\} \in TL(L, S)$ □

The following are definitions of the languages, except for AL which is defined in Def. 3.5. Note that AL_T was called TL in Chapter 3 (the more general naming convention is used here for clarity).

All the following definitions are implicitly parameterized by a number n and a set Θ of primitive propositions, and all except AL and AL_T in addition by two sets V_F and V_T of variables. For brevity I drop a notation with explicit parameters.

Definition 7.2 (Term Languages)

- $AL_T \equiv TL(AL)$
- $AL_{VT} \equiv TL(AL_V, V_T)$
- $TRL_T \equiv TL(TRL)$ □

Definition 7.3 (AL_V,TRL) AL_V is the least set such that:

- $\Theta \cup V_F \subseteq AL_V$
- If $T^V, U^V \in AL_{VT}$ then $T^V \doteq U^V \in AL_V$
- If $T^V \in AL_{VT}$ then
  - $\triangle_i T^V \in AL_V$
  - $\triangledown_i T^V \in AL_V$
- If $\alpha^V, \beta^V \in AL_V$ then
  - $(\alpha^V \to \beta^V) \in AL_V$
  - $\neg\alpha^V \in AL_V$

TRL is the least set such that:

- If $T^V, U^V \in AL_{VT}$ then $\frac{T^V}{U^V} \in TRL$ □

Definition 7.4 (TEL) TEL is the least set such that:

- $\Theta \subseteq TEL$

[2] $TL(L) = TL(L, \emptyset)$.

- If $T, U \in AL_T$ then $T \doteq U \in TEL$
- If $T \in AL_T$ then
  - $\triangle_i T \in TEL$
  - $\triangledown_i T \in TEL$
- If $T^R \in TRL_T$ then
  - $\overset{\leadsto}{\triangle}_i T^R \in TEL$
  - $\overset{\leadsto}{\triangledown}_i T^R \in TEL$
- If $\alpha, \beta \in TEL$ then
  - $(\alpha \to \beta) \in TEL$
  - $\neg\alpha \in TEL$
- If $\alpha, \beta \in TEL$ and $G \in \wp(\{1, \ldots, n\})$ then
  - $\langle\langle A \rangle\rangle \bigcirc \alpha \in TEL$
  - $\langle\langle A \rangle\rangle \Box \alpha \in TEL$
  - $\langle\langle A \rangle\rangle \alpha \mathcal{U} \beta \in TEL$ □

Note that $\Theta \subset AL \subset AL_V \supset V_F$ and $AL_T \subset AL_{VT} \supset V_T$.

As in Part II the following notation will be used for meta variables: $\phi, \psi, \ldots$ for elements in TEL, $\alpha, \beta, \ldots$ for elements in AL and $T, U, \ldots$ for elements in AL_T. In addition, $\alpha^V, \beta^V, \ldots$ will be used for elements in AL_V, $T^V, U^V, \ldots$ for elements in AL_{VT}, $R_1, R_2, \ldots$ for elements in TRL, $T_1^R, T_2^R, \ldots$ for elements in TRL_T, $a, b, \ldots$ for elements in V_F and $t, u, \ldots$ for elements in V_T.

7.2.5 Substitutions

A *substitution* Ω is a pair of functions

$$\Omega_F : V_F \to AL$$
$$\Omega_T : V_T \to AL_T$$

mapping each formula-variable $a \in V_F$ to an agent formula and each term-variable $t \in V_T$ to an term for a set of agent formulae. The set of all substitutions is called *Subst*. In the following, we abuse notation and write only Ω for both Ω_F and Ω_T. Sometimes the notation

$$\Omega[x/y]$$

is used, where $x \in V_F$ and $y \in AL$, or $x \in V_T$ and $y \in AL_T$, to denote the substitution Ω' where:

$$\Omega'(x') = \begin{cases} y & x = x' \\ \Omega(x') & \text{otherwise} \end{cases}$$

Substitutions will be used to define *instances* of a rule $R \in TRL$ — i.e. to identify agent formulae *matching* the antecedent and consequent of the rule.

Definition 7.5 (T_Ω^V) Let $\Omega \in Subst$ and $T^V \in AL_{VT}$. T_Ω^V is the result of replacing every variable $x \in V_F \cup V_T$ occurring in T^V with $\Omega(x)$. Clearly, $T_\Omega^V \in AL_T$. □

7.3 Interpretation of Rules

In order to define a semantical interpretation of rules, first the interpretation of a syntactical rule term as a set of rules is defined, in the same manner as for terms of formulae in Chapter 3.

Definition 7.6 (Interpretation of Rule Terms) The interpretation $[T^R] \in \wp^{fin}(TRL)$ of a rule term $T^R \in TRL_T$ is defined by Def. 3.3 by taking $L = S = TRL$ and $[] : TRL \to TRL$ to be the identity function. □

The semantics of a rule term is defined by combining all substitutions with all rules in the interpretation of the rule term.

Definition 7.7 ($[\![T^R]\!]$) Let $T^R \in TRL_T$ be a rule term. $[\![T^R]\!]$ is the following relation:

$$[\![T^R]\!] = \{([T^A_\Omega], [T^C_\Omega]) : \frac{T^A}{T^C} \in [T^R], \Omega \in Subst\}$$

I will write $[\![T^R]\!](s)$ for $\{s' : (s, s') \in [\![T^R]\!]\}$. □

In the above definition, $T^A, T^C \in AL_{VT}$, $T^A_\Omega, T^C_\Omega \in AL_T$ and $[T^A_\Omega], [T^C_\Omega] \subseteq \wp^{fin}(OL)$.

Example 7.1

$$[\![\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{b\}}]\!](s) = \{s' \cup \{\beta\} : s = s' \cup \{\alpha, \alpha \to \beta\}, s' \in \wp^{fin}(OL), \alpha, \beta \in OL\}$$

$$[\![\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{a, a \to b, b\}}]\!](s) = \{s \cup \{\beta\} : \alpha, \alpha \to \beta \in s\}$$

Note an important property of the $\sqcup$ operator: it denotes not necessarily disjoint union. Thus, the variable t in the first rule *can* denote a set which includes the premises denoted by the expressions a and $a \to b$ in the premise. These two rules will be discussed further in Section 7.6.1. □

Informally, the intended meaning of "knowing rules T^R" is to have a mechanism which in a state s, for each $s' \in [\![T^R]\!](s)$, can make the next state, for the agent itself in case of reasoning or for another agent in case of communication, to be at or above s'. Before defining this formally in Section 7.5, a closer look at concurrent game structures is taken.

7.4 Concurrent Game Structures and Strategies

The ATL framework (Section 2.3) is chosen as a framework for the logic and concurrent game structures will be used to model the temporal dimension of the agent system. Before the model is formally presented, let us briefly look at some properties of the concurrent game structures which we are interested in.

First, we require the ("global") states to be a composition of k "local" states – one epistemic state (or point in the $\wp^{fin}(OL)$ lattice) for each agent. This is the first condition for Moore synchronous game structures (see Section 2.3.3). Second, we require that the number of possible moves for an agent is a function of the agent's *local* state. An agent's mechanism can allow it to make several decisions based on its local state, but the possible decisions do not depend on the local states of the other agents. Formally $d_a(q_1, \ldots, q_k) = d_a(q'_1, \ldots, q'_k)$ whenever $q_a = q'_a$. We do not in general require that the second condition for Moore synchronous game structures holds because we will allow communication between agents, but we do consider the special case without communication, where precisely this condition holds, in Section 7.7. Third, we require a specific state space for the local states, namely $\wp^{fin}(OL)$. Fourth, we require a fixed δ. This δ determines how the agents' decisions about how to use their mechanisms transform one tuple of local states to another. The actual function δ is discussed below.

These four requirements define the subset of concurrent game structures that will be uses to model the agent systems.

In addition, we must restrict the notion of a strategy. In ATL a strategy is defined as a function mapping a history of global states to a decision. This definition is too general for two reasons. First, an agent cannot discern between two global states in which (only) the epistemic states of other agents differ. Second, in order to take seriously the idea of an epistemic state as "everything an agent knows", we must let the agent's strategy depend upon only the *current* state rather than the *history* of states. If we want the agent to be able to model recollection of past states (e.g. perfect recall, a common assumption in game theory), we must encode that knowledge in the agent language. Thus, a valid strategy maps a local state to one of the possible decisions in that state.

In addition to the specializations just discussed, a slight generalization of the concept of concurrent game structures, as defined in Section 2.3.1, is needed. The reason is that in the original definition, concurrent game structures are *finitely restricted* in several ways[3]. For example, the cardinality of the state space is required to be finite. These restrictions do not hold, however, in the model we want to use of transitions between epistemic states where e.g., although the epistemic states themselves are required to be finite, the number of possible epistemic states is infinite.

7.4.1 Concurrent Game Structures Generalized

The concept (Def. 2.1) of a concurrent game structure is redefined as follows.

Definition 7.8 A concurrent game structure is a tuple

$$(k, Q, \Pi, \pi', ACT, d, \delta)$$

where

- $k > 0$ is a natural number of *players*
- Q is a set of *states*
- Π is a set of *propositions*
- $\pi'(q) \subseteq \Pi$ for each $q \in Q$; *the labeling function*

[3]One reason for these restrictions in the original definition of concurrent game structures is that ATL is presented, and often used, for model checking. The restrictions ensure decidability and desirable complexity properties of model checking.

- ACT is a set of *actions*
- For each player $a \in \{1, \ldots, k\}$ and state $q \in Q$, $d_a(q) \subseteq ACT$ is the set of *moves* available to player a in q. $D(q) = d_1(q) \times \cdots \times d_n(q)$ is the set of *move vectors* in q.
- For each move vector $v \in D(q)$ in a state $q \in Q$, $\delta(q, v) \in Q$; the *transition function*. □

The following are changed from Def. 2.1. Q and Π can be infinite in order to allow for infinite state spaces and infinitely many primitive propositions, respectively. The function d_a is changed to map to a general set of actions ACT. The only requirement on ACT is that it is a set. The reason for this change is twofold. First, it makes it easy to define the concept of choosing the *same action* in two different states by simply comparing the action names. This can also be done with action numbers, but the assumption that actions with the same numbers in different states should be the same action seems less natural[4]. Second, it allows us to model a situation where an agent has an infinite number of available actions. This will be the case in the situations we want to model here.

The concept of a strategy is redefined accordingly: a strategy for player i is a function $f_i : Q^+ \to ACT$ where $f_i(q_0 \cdots q_m) \in d_i(q_m)$.

The concepts of a computation and a set of strategy vectors, and the function *out*, are not changed (see Section 2.3.1). Henceforth, the notions "concurrent game structure" and "strategy" will refer to the new definitions here, unless otherwise noted. The definition of satisfiability of an ATL formula ϕ in state q of a concurrent game structure S,

$$S, q \models \phi$$

remains the same (Def. 2.2) despite the slight change in the definitions of concurrent game structures and strategies.

7.4.2 Concurrent Game Structures with Incomplete Information

Clearly, in the concurrent game structures of interest, the players have *incomplete information* about the current (global) state. Two natural restrictions on such structures, and their associated strategies, are the following:

1. An agent must have the same actions available in two states which are indiscernible for that agent.
2. A strategy must map indiscernible histories of states to the same action.

In the paper presenting ATL, Alur, Henzinger, & Kupferman (1997) propose a definition, including the two mentioned restrictions, of game structures with incomplete information only in the turn-based synchronous case (see Section 2.3.3). Jamroga (2003) generalizes this idea to AETSs (see Section 9.4.1); the new definition corresponds exactly to the two restrictions above. The following is a further generalization, of a general requirement of concurrent game structures with incomplete information.

[4]The use of a set ACT is suggested by Jamroga (2003) for among others this reason.

Definition 7.9 A *game structure with incomplete information* is a concurrent game structure $(k, Q, \Pi, \pi, ACT, d, \delta)$ together with a set $\Pi_a \subseteq \Pi$ for each player a such that

$$d_a(q) = d_a(q')$$

for any states $q, q' \in Q$ such that

$$\pi_a(q) = \pi_a(q')$$

where we write $\pi_a(q'') = \pi(q'') \cap \Pi_a$ for any $q'' \in Q$. □

$\pi_a(q)$ is the set of formulae which is both true in q and observable by a.

A *strategy* for a player a in a concurrent game structure with incomplete information is a strategy with the following restriction:

$$f_a(q_1 \cdots q_m) = f_a(q'_1 \cdots q'_m)$$

for any $m \geq 0$ and $q_i \in Q$ having

$$\pi_a(q_i) = \pi_a(q'_i)$$

for all $1 \leq i \leq m$. Thus, for satisfiability with respect to a concurrent game structure with incomplete information only these restricted strategies will be considered.

Clearly, the definitions are very general adaptions of the turn-based case.

Turn-based synchronous game structures with incomplete information are also restricted so that an agent can only influence observable propositions, and semantics are only defined for a syntactically restricted subset of the ATL language (see Section 2.3.3). It is not clear, however, that these restrictions are necessary. Consider a case with two players, 1 and 2, where p stands for the proposition "player 1 knows that player 1's hat is white". Intuitively, p should not be considered as an observable proposition for player 2. Nevertheless, if we model a system in which there are communication actions, a formula such as $\phi = \langle\langle 2 \rangle\rangle \bigcirc p$, which would be left out of the restricted language just mentioned, makes sense[5]. Thus, I disagree with the syntactical restriction and will hence consider the semantics of the full ATL language with respect to concurrent game structures with incomplete information.

Non-perfect Recall

The above definitions implicitly assume, similarly to previous proposals for modelling incomplete information in the ATL framework, that agents have perfect recall, by allowing a strategy to map indiscernible states with different histories to different actions.

Non-perfect recall can be modelled by further restricting the strategies. For satisfiability of an ATL formula in a concurrent game structure with incomplete information *without perfect recall*, the set of strategies is restricted as follows:

$$f_a(q_1 \cdots q_m) = f_a(q'_1 \cdots q'_o)$$

[5]It can be argued that p *becomes* observable by player 2 when he makes the communication action. I do not agree with this either; if player 2 does not know that the communication is reliable he may not know whether his action was successful. The formula still makes sense.

for any $m, o \geq 0$ having
$$\pi_a(q_m) = \pi_a(q'_o)$$

In the next section, I will describe a class of concurrent game structures with incomplete information and use them in the context of non-perfect recall.

7.5 Mechanisms

A *mechanism* is a model of an agent's reasoning and communication abilities.

Definition 7.10 (Mechanism) A *mechanism* (for n agents) is a tuple
$$R = (R_1, \ldots, R_n)$$
where
$$R_i \subseteq \wp^{fin}(OL) \times (\wp^{fin}(OL))^n$$
and
$$R_i(s) = \{g : (s, g) \in R_i\} \neq \emptyset$$
for every $s \in \wp^{fin}(OL)$. □

We write $R_i(s)^j$ for the set of j-projections of the tuples $R_i(s)$:
$$R_i(s)^j = \{s_j : (s_1, \ldots, s_n) \in R_i(s)\}$$

Agent i having mechanism R_i is intended to model the fact that in a state of the system where agent i has local epistemic state s_i, he can force the system into a new state in which each agent j (including i) is at or above, depending on the other agents' mechanisms and decisions, the state s'_j where $R_i(s_i) = (s'_1, \ldots, s'_n)$ in the lattice of possible epistemic states. Note that it is required that $R_i(s)$ always assigns a tuple to a state, it can e.g. be the case that $R_i(s)$ assigns s to agent i – meaning "no change".

The semantics of the temporal parts of the language is defined in relation to a mechanism, in terms of the concurrent game structure which is induced by the mechanism.

7.5.1 Induced Concurrent Game Structure

In Chapter 3 semantics of the language *EL* was defined in relation to a set of points and a truth assignment of Θ. The new language *TEL* is an extension of *EL* with rule operators and temporal connectives, and the semantics can be extended for these new elements by adding a mechanism. The mechanism describes possible future states, while the states themselves describe the semantics of the *EL* subset of the language. This semantics can be easily described by a concurrent game structure.

Definition 7.11 (Ind. Conc. Game Structure with Incomplete Inform.) Given a mechanism for n agents, $R = (R_1, \ldots, R_n)$, and a truth assignment $\pi : \Theta \rightarrow \{\textbf{true}, \textbf{false}\}$ of Θ, the *induced concurrent game structure with incomplete information* is the concurrent game structure
$$\overrightarrow{R}_\pi = (n, Q, \Pi, \pi', ACT, d, \delta)$$

together with sets Π_a of observable propositions for each agent $1 \leq a \leq n$ where

- $Q = (\wp^{fin}(OL))^n$
- $\Pi = \Theta$
  $\cup \{T \doteq U : T, U \in AL_T\}$
  $\cup \{\triangle_i T, \bigtriangledown_i T : T \in AL_T, 1 \leq i \leq n\}$
  $\cup \{\overset{\leadsto}{\triangle}_i T^R, \overset{\leadsto}{\bigtriangledown}_i T^R : T^R \in TRL_T, 1 \leq i \leq n\}$
- Let $q = (s_1, \ldots, s_n) \in Q$. Let $\phi \in \Pi$.
  - $\phi = p \in \Theta$: $\phi \in \pi'(q)$ iff $\pi(p) = \mathbf{true}$.
  - $\phi = T \doteq U$: $\phi \in \pi'(q)$ iff $[T] = [U]$.
  - $\phi = \triangle_i T$: $\phi \in \pi'(q)$ iff $[T] \subseteq s_i$.
  - $\phi = \bigtriangledown_i T$: $\phi \in \pi'(q)$ iff $s_i \subseteq [T]$.
  - $\phi = \overset{\leadsto}{\triangle}_{ij} T^R$: $\phi \in \pi'(q)$ iff

$$s' \in [\![T^R]\!](s_i) \text{ and } (s'_1, \ldots, s'_n) \in R_i(s_i)$$
$$\Downarrow$$
$$(s''_1, \ldots, s''_n) \in R_i(s_i) \text{ where } s''_k = \begin{cases} s' & k = j \\ s'_k & \text{otherwise} \end{cases}$$

  - $\phi = \overset{\leadsto}{\bigtriangledown}_{ij} T^R$: $\phi \in \pi'(q)$ iff $(s'_1, \ldots, s'_n) \in R_i(s_i) \Rightarrow s'_j \in [\![T^R]\!](s_i)$
- $ACT = (\wp^{fin}(OL))^n$
- Let $q = (s_1, \ldots, s_n) \in Q$ and $1 \leq a \leq n$. $d_a(q) = R_a(s_a) \subseteq ACT$.
- Let $q = (s_1, \ldots, s_n) \in Q$ and $v = (g_1, \ldots, g_n) \in D(q) = R_1(s_1) \times \cdots \times R_n(s_n)$ where $g_i = (s^i_1, \ldots, s^i_n)$. Then,

$$\delta(q, v) = (\bigcup_{i=1}^{n} s^i_1, \ldots, \bigcup_{i=1}^{n} s^i_n)$$

- Let $i \in [1, n]$. $\Pi_i = \{\triangle_i T, \bigtriangledown_i T : T \in AL_T\}$ □

The set of actions available in a state, $d_a(q)$, is the set described by a's mechanism. π' extends the truth assignment of Θ to a truth assignment for all the EL-formulae, in addition to the new rule formulae, relative to a state. The rule formulae are also evaluated in a state, but relative to the mechanism. The semantics of rule formulae is discussed in detail in the next section, but note that knowing "at least" a rule for communication to an agent requires that the rule can be used together with all other possibilities for communication with the other agents. As discussed in Section 7.4, the set of possible decisions and the choice in a strategy should not depend on the other agents' states – incomplete information. δ maps an action $(s^i_1, \ldots, s^i_n)$ for each agent i to a new tuple of epistemic states $(\cup_{i=1}^n s^i_1, \ldots, \cup_{i=1}^n s^i_n)$. In other words, that agent i uses action $(s^i_1, \ldots, s^i_n)$ means that he "sends" s^i_j to each agent j – including himself – and thereby forces the new epistemic state of j to be at or above s^i_j. Observable propositions are exactly those which describe the agent's epistemic states:

Lemma 7.1 If $\overrightarrow{R}_\pi = (n, Q, \Pi, \pi', ACT, d, \delta)$ is an induced concurrent game structure, then ($\pi_a(s) = \Pi_a \cap \pi'(s)$):

$$\pi_a(s_1, \ldots, s_n) = \pi_a(s'_1, \ldots, s'_n) \Leftrightarrow s_a = s'_a \qquad \Box$$

PROOF If $\pi_a(s_1, \ldots, s_n) = \pi_a(s'_1, \ldots, s'_n)$, let $T \in AL_T$ be such that $[T] = s_a$. $\triangle_a T, \triangledown_a T \in \pi_a(s_1, \ldots, s_n)$, so $\triangle_a T, \triangledown_a T \in \pi_a(s'_1, \ldots, s'_n)$ and thus $s'_a = [T] = s_a$. The other direction follows directly from the definition. ■

It is easy to see that Def. 7.11 is proper:

Lemma 7.2 If R is a mechanism, then $\overrightarrow{R}_\pi$ is a concurrent game structure with incomplete information. □

PROOF It is easy to see from the definition that $\overrightarrow{R}_\pi$ is a concurrent game structure. Let $\pi_a(q) = \pi_a(q')$. For incomplete information, it must be the case that $d_a(q) = d_a(q')$, i.e. that $R_a(s_a) = R_a(s'_a)$ where $q = (s_1, \ldots, s_n)$ and $q' = (s'_1, \ldots, s'_n)$, which follows directly from Lemma 7.1. ■

Since the induced concurrent game structure is defined with incomplete information, the notion of a strategy will be restricted accordingly (see Sec. 7.4.2).

A *computation* λ in the induced concurrent game structure is a sequence of states where for every $j \geq 0$ $\lambda[j+1] = (\cup_{i=1}^n s_1^i, \ldots, \cup_{i=1}^n s_n^i)$ where $(s_1^i, \ldots, s_n^i) \in R_i(\lambda[j]^i)$ for all $i \in [1, n]$, where $\lambda[j]^i$ is the jth component of $\lambda[j]$. A computation in $\overrightarrow{R}_\pi$ is called a R-computation.

The notion of truth for the different parts of the language is discussed in further details in the following section.

7.6 Satisfiability

TEL can be seen as an ATL language over the set of propositions Π defined in the construction of the induced concurrent game structure in the previous section, and satisfiability of a formula in a mechanism can thus be defined as satisfiability of the formula in the induced concurrent game structure, as defined in Section 2.3.2. Note that the game structure has incomplete information, and that we also restrict the strategies further by not allowing perfect recall.

The logic, in a broad use of the concept, over the language *TEL* defined by the satisfiability definition below is called DSEL — *Dynamic Syntactic Epistemic Logic*.

Definition 7.12 (Satisfiability of *TEL*) A formula $\phi \in TEL$ is *satisfied* by a mechanism R and a KSS $M = (s_1, \ldots, s_n, \pi) \in \mathcal{M}_{fin}$ written

$$R, M \models \phi$$

iff

$$\overrightarrow{R}_\pi, (s_1, \ldots, s_n) \models \phi$$

without perfect recall. □

As usual,

$$R \models \phi$$

means that $R, M \models \phi$ for all M. For brevity, a tuple $(s_1, \ldots, s_n)$ will sometimes be written $\vec{s}$. In addition, the jth component of a tuple will sometimes be referenced by using j as a superscript. For example, if λ is a computation then $\lambda[k]^j = s_j$ where $\lambda[k] = (s_1, \ldots, s_n) \in (\wp^{fin}(OL))^n$.

The reason for using the induced concurrent game structure without perfect recall is, as discussed in Section 7.4, that we want all the agents' knowledge to be encoded in their epistemic states.

In the definition of satisfiability of *TEL* strategies are restricted by incomplete information and non-perfect recall. A strategy is a function $f_i : Q^+ \rightarrow ACT$ with $f_i(q_0, \ldots, q_m) \in d_i(q_m)$ and $f_i(q_0, \ldots, q_m) = f_i(q'_0, \ldots, q'_m)$ whenever $\pi_i(q_m) = \pi_i(q'_m)$ which is exactly when i has the same state in q_m and q'_m (Lemma 7.1). A strategy can thus be written as a function

$$f_i : \wp^{fin}(OL) \rightarrow (\wp^{fin}(OL))^n$$

having $f_i(s) \in R_i(s)$. Henceforth we will use this shorthand definition of a strategy. Strategies are defined for a particular concurrent game structure. This will sometimes be made explicit by using the notation $Str(G, R)$ for the set of strategies for agents G in $\overrightarrow{R}_\pi$ (arbitrary π, strategies do not depend on π); $Str(G)$ is used when no confusion can occur.

Similarly, the function *out* is also defined for a particular structure. Let G be a set of agents, $R = (R_1, \ldots, R_n)$ a mechanism, $\vec{s} \in (\wp^{fin}(OL))^n$, and $\vec{f}_G \in Str(G, R)$. The explicit notation $out_R(\vec{f}_G, \vec{s})$, will sometimes be used[6]. It is easy to see (see Section 2.3.1) that $\lambda \in out(\vec{f}_G, \vec{s})$ iff

1. $\lambda[0] = \vec{s}$
2. $\forall_{j \geq 0} \lambda[j+1] = (\cup_{i=1}^n s_1^i, \ldots, \cup_{i=1}^n s_n^i)$ where
   - For all $i \in [1, n]$: $(s_1^i, \ldots, s_n^i) \in R_i(\lambda[j]^i)$
   - For all $i \in G$: $(s_1^i, \ldots, s_n^i) = f_i(\lambda[j]^i)$

Note the special situation when $G = \emptyset$: $Str(\emptyset, R) = \vec{f}_\emptyset$ (the unique empty strategy vector), $out(\vec{f}_\emptyset, \vec{s})$ is the set of all computations λ with $\lambda[0] = \vec{s}$, and $R, (\vec{s}, \pi) \models \langle\langle\emptyset\rangle\rangle\Box\phi$ implies that $R, (\lambda[k], \pi) \models \phi$ for any k and any R-computations λ with $\lambda[0] = \vec{s}$. Also, $R \models \phi \Rightarrow R \models \langle\langle\emptyset\rangle\rangle\Box\phi$.

The induced concurrent game structure provides semantics for *EL* formulae and for formulae starting with the new rule operators via π'; the latter is discussed in further details in the following subsection. For formulae starting with the new temporal operators, the semantics are defined for the corresponding ATL formulae in terms of strategies and computations.

An example is presented after discussions about the relation between mechanisms and rules, and between rules and temporal properties.

[6]In the definition of satisfiability of ATL formulae in a concurrent game structure (Section 2.3.2) the notation $Str(G)$ and $out(q, \vec{f})$ is used without explicit reference to the structure. Of course, for satisfiability in an induced concurrent game structure $\overrightarrow{R}_\pi$, $Str(G) = Str(G, R)$ and $out(q, \vec{f}) = out_R(q, \vec{f})$.

7.6.1 Mechanisms and Rules

Rules, together with the rule operators, express properties of a mechanism. Such properties are investigated in this section.

As already mentioned, the expressions for knowing (at least or at most) a set of formulae and knowing (at least or at most) a set of rules are quite similar. The two concepts they denote, however, are fundamentally different. Knowing a set of formulae is a *syntactic* notion, while knowing a set of rules is a *semantic* notion. The former is defined as membership in the agent's (syntactic) epistemic state, while the latter is defined as a property of the mechanism. An agent does not have a syntactic representation of rules; it only has a syntactic representation of formulae together with a mechanism for manipulating them. Consider first rules used for reasoning (as opposed to communication). "Agent j knows at least Modus Ponens" can be expressed by the formula

$$\phi_1 = \overset{\rightsquigarrow}{\triangle}_{jj}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{a, a \to b, b\}}\}$$

If this formula is true and $p \to q, p$ is in j's epistemic state s_j, then j's mechanism can assign a new epistemic state to j which is s_j extended with q ($s_j \cup \{q\} \in R_j(s_j)^j$) (see Example 7.1 on p. 88). Moreover, the mechanism can assign this state to the agent *independently* of the states it assigns to other agents. Note that the assigned state is not necessarily the new epistemic state; the latter can also be influenced by communication as will be discussed below. A slightly different version of ϕ_1 is:

$$\phi_2 = \overset{\rightsquigarrow}{\triangle}_{jj}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{b\}}\}$$

(again, see Example 7.1). With the new rule, j's mechanism can still produce the state $s_j \cup \{q\}$ – but it can also produce e.g. $(s_j \setminus \{p\}) \cup \{q\}$. In other words, the agent may forget one of the modus ponens premises, both of them, or none of them.

An agent will in the general case know infinitely many rules. For example, he will know every rule with an antecedent which does not match the agent's epistemic state. Also, knowing a rule may imply knowing several more specific rules. For example,

$$\models \phi_2 \to \phi_1$$

(where ϕ_1 and ϕ_2 are as defined above). That is, if a mechanism can do all the things expressed by ϕ_2 in a certain epistemic state, it can also do all the things expressed by ϕ_1 in the same state.

Rules used for communication work in exactly the same way: agent i knowing the rule "if I know the formula p then I can tell it to agent j" is expressed as

$$\phi_3 = \overset{\rightsquigarrow}{\triangle}_{ij}\{\frac{t \sqcup \{p\}}{\{p\}}\}$$

This rule requires that i's mechanism can assign the partial state $\{p\}$ to agent j. Although there is no significant semantical difference between knowing a rule for reasoning and knowing a rule for communication, the use of them will typically be different. Rules for reasoning will typically be *monotone*; the consequent will preserve (parts of) the antecedent. Monotonicity is discussed below.

A rule such as the one in ϕ_3 will make more sense as a rule for communication than for reasoning, while the reasoning rules in ϕ_1 and ϕ_2 will not be typical communication rules. The rule in ϕ_3 used as a reasoning rule would loose all information except p; the rules in ϕ_1 and ϕ_2 used as communication rules would send the entire epistemic state to the other agent. If an agent knows rules for both reasoning and for communication, the mechanism must be able to produce all combinations of the rule applications (more on this below).

Knowing *at least* a rule means to be able to use the rule in all possible "ways". Knowing *at most* a rule means that everything that can be done with the mechanism is an application of the rule (but it is not necessary that all applications of the rule can be used). For example, the formula

$$\phi_4 = \overset{\leadsto}{\nabla}_{ij}\{\frac{t}{\triangle_i\{a\}}\}$$

means that agent i can only communicate facts on the form "I know that α", where α is a formula in i's epistemic state, to agent j. Of course, the argument to both types of rule operators is a term, so the following is well formed:

$$\phi_5 = \overset{\leadsto}{\nabla}_{ij}\{\frac{t}{\triangle_i\{a\}}, \frac{t}{\nabla_i u}\}$$

expressing the fact that i can only tell j "I know that α" *or* "I know at most X", where α/X is a formula/finite set of formulae in i's epistemic state. Note that ∇ijT^R with a non-singular rule term T^R, as in ϕ_5, means that all that i can communicate to j is described by the "union" of all the rules in the rule term.

As for the epistemic operators, a derived rule operator is defined:

$$\overset{\leadsto}{\Diamond}_{ij}T^R \equiv \overset{\leadsto}{\triangle}_{ij}T^R \wedge \overset{\leadsto}{\nabla}_{ij}T^R$$

Note that $\overset{\leadsto}{\Diamond}_{ij}T^R$ does not necessarily mean that i knows only the rules T^R. For example, as discussed above,

$$\models \overset{\leadsto}{\Diamond}_{jj}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{b\}}\} \to \phi_1$$

(where ϕ_1 is as defined above).

The following Lemma states some results about the relation between knowing rules and having a mechanism.

Lemma 7.3 Let $R = (R_1, \ldots, R_n)$ be a mechanism, $M = (s_1, \ldots, s_n, \pi) \in \mathcal{M}_{fin}$ and $T^R, T_1^R, \ldots, T_n^R \in TRL_T$.

1. $(R, M \models \overset{\leadsto}{\triangle}_{ij}T^R) \Rightarrow [\![T^R]\!](s_i) \subseteq R_i(s_i)^j$.

2. $(R, M \models \overset{\leadsto}{\nabla}_{ij}T^R) \Leftrightarrow R_i(s_i)^j \subseteq [\![T^R]\!](s_i)$.

3. $(R, M \models \overset{\leadsto}{\Diamond}_{ij}T^R) \Rightarrow R_i(s_i)^j = [\![T^R]\!](s_i)$.

4. $(R, M \models \bigwedge_{j \in [1,n]} \overset{\leadsto}{\Diamond}_{ij}T_j^R) \Leftrightarrow R_i(s_i) = [\![T_1^R]\!](s_i) \times \cdots \times [\![T_n^R]\!](s_i)$.

5. If $R_i(s_i) = X_1 \times \cdots \times X_n$ where $X_j = [\![T^R]\!](s_i)$ then $R, M \models \overset{\rightsquigarrow}{\Diamond}_{ij} T^R$ $\square$

PROOF

1. Let $R, M \models \overset{\rightsquigarrow}{\triangle}_{ij} T^R$, and let $s' \in [\![T^R]\!](s_i)$. Since $R_i(s_i) \neq \emptyset$ (by definition), there is an element $(s'_1, \ldots, s'_n) \in R_i(s_i)$. By the definition of satisfiability of $\overset{\rightsquigarrow}{\triangle}_{ij} T^R$, there must be a $(s''_1, \ldots, s''_n) \in R_i(s_i)$ such that $s''_j = s'$. Thus, $s' \in R_i(s_i)^j$.

2. Follows immediately from the definition of satisfiability of $\overset{\rightsquigarrow}{\triangledown}_{ij} T^R$.

3. Follows immediately from 1. and 2.

4. **⇒)** Let $R, M \models \bigwedge_{j \in [1,n]} \overset{\rightsquigarrow}{\Diamond}_{ij} T^R_j$.
   - **⊆)** Let $(s'_1, \ldots, s'_n) \in R_i(s_i)$. By 2. for each j, $s'_j \in [\![T^R_j]\!](s_i)$.
   - **⊇)** Let $(s'_1, \ldots, s'_n) \in [\![T^R_1]\!](s_i) \times \cdots \times [\![T^R_n]\!](s_i)$. By definition, there is a $(s''_1, \ldots, s''_n) \in R_i(s_i)$. By the definition of sat. of $\overset{\rightsquigarrow}{\triangle}_{ij}$, $(s'_1, s''_2, \ldots, s''_n) \in R_i(s_i)$, and so on, and $(s'_1, \ldots, s'_n) \in R_i(s_i)$ (by repeating the argument n times).

   ⇐) Let $R_i(s_i) = [\![T^R_1]\!](s_i) \times \cdots \times [\![T^R_n]\!](s_i)$, and let $j \in [1, n]$. If $s' \in [\![T^R_j]\!](s_i)$ and $(s'_1, \ldots, s'_n) \in R_i(s_i)$, then also $(s'_1, \ldots, s'_n)$ with s'_j replaced by s' must be in $R_i(s_i)$. Thus, $R, M \models \overset{\rightsquigarrow}{\triangle}_{ij} T^R_j$ for any j. For every $(s'_1, \ldots, s'_n) \in R_i(s_i)$, $s'_j \in [\![T^R_j]\!](s_i)$ for every j, so $R, M \models \overset{\rightsquigarrow}{\triangledown}_{ij} T^R_j$ for any j.

5. Follows immediately from the definition of satisfiability of $\overset{\rightsquigarrow}{\triangle}_{ij} T^R$ and $\overset{\rightsquigarrow}{\triangledown}_{ij} T^R$. ■

Lemma 7.3 shows that if agents are described with $\overset{\rightsquigarrow}{\triangle}/\overset{\rightsquigarrow}{\triangledown}$, we can determine a minimal/maximal mechanism satisfying the description — provided that it exists. Particularly, lemma 7.3.4 shows that certain mechanisms can be completely described by sets of rules. Mechanisms are *more general* than rules: knowing exactly a set of rules corresponds to having a certain mechanism, but having a certain mechanism does not necessarily correspond to knowing a set of rules. Mechanisms are more general because they allow a state to map to *any* set of tuples, while in a mechanism completely described by a set of rules a state must map to a specific Cartesian product (Lemma 7.3.4). A consequence of this is a restriction in the expressiveness of the language: we can express exactly the possibilities an agent has to communicate to another agent in a certain state, but we cannot express restrictions on the relation between what an agent sends to one agent and to another agent. For example, if agent i knows one rule for communicating to agent j and another for communicating to agent k, the agent will necessarily have the ability to use the two rules *simultaneously*.

Note that the other direction of Lemma 7.3.1 does not hold, and thus neither does point 3 of the same Lemma. As a counter example, consider a case where

$n = 2, \Theta = \{p, q\}, s_1 = \{p\}, T^R = \{\frac{p}{q}\}$ and $R_1(s_1) = \{(\{p\}, \{q\}), (\{q\}, \{p\})\}$. $[\![T^R]\!](s_1) = \{\{q\}\} \subseteq R_1(s_1)^2 = \{\{q\}, \{p\}\}$, but $R, M \not\models \overset{\rightsquigarrow}{\triangle}_{12} T^R$ because $\{q\} \in [\![T^R]\!](s_1)$ and $(\{q\}, \{p\}) \in R_1(s_1)$ but $(\{q\}, \{q\}) \notin R_1(s_1)$.

Of course, not all sets of rules can be used to describe mechanisms. Those which can are called *complete*.

Definition 7.13 (Complete rules) A rule term T^R is *complete* iff for all $s \in \wp^{fin}(OL)$, $[\![T^R]\!](s) \neq \emptyset$. □

Lemma 7.4 Let $T_1^R, \dots, T_n^R \in TRL_T$ and $i \in [1, n]$. $T_1^R, \dots, T_n^R$ are complete iff there exists a mechanism R such that

$$R \models \bigwedge_{j \in [1,n]} \overset{\rightsquigarrow}{\Diamond}_{ij} T_j^R$$

□

Proof Let $T_1^R, \dots, T_n^R$ be complete and let $R = (R_1, \dots, R_n)$ be such that for any $s \in \wp^{fin}(OL)$, $R_i(s) = [\![T_1^R]\!](s) \times \cdots \times [\![T_n^R]\!](s)$ and $R_j(s)$, $j \neq i$, is arbitrary but non-empty. Since $[\![T_j^R]\!](s)$ are non-empty for each j, $R_i(s)$ is also non-empty and thus R is a mechanism. For any $M = (s_1, \dots, s_n, \pi)$, $R, M \models \bigwedge_{j \in [1,n]} \overset{\rightsquigarrow}{\Diamond}_{ij} T_j^R$ by Lemma 7.3.4. The other direction follows by the same Lemma; since $R_i(s) = [\![T_1^R]\!](s) \times \cdots \times [\![T_n^R]\!](s)$ for any s and $R_i(s) \neq \emptyset$, $[\![T_j^R]\!](s) \neq \emptyset$ for any s. ■

Intuitively, a rule can be used in several different ways, depending on the rule and the current epistemic state. Deterministic rules can always be used in only one way.

Definition 7.14 (Deterministic rules) A rule term $T^R \in TRL_T$ is *deterministic* iff $[\![T^R]\!](s)$ is singular for every s. □

The following example illustrates the use of rules.

Example 7.2 The formula in the example in Section 7.2.3 is valid:

$$\begin{aligned} \models (&\triangle_1\{p\} \wedge \triangle_2\{p \to q\} \\ &\wedge \langle\langle \emptyset \rangle\rangle \Box \left(\overset{\rightsquigarrow}{\triangle}_{12}\{\frac{t \sqcup \{a\}}{\{a\}}\} \wedge \overset{\rightsquigarrow}{\triangle}_{22}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{a, a \to b, b\}}\} \wedge \overset{\rightsquigarrow}{\triangle}_{22}\{\frac{t}{t}\} \right)) \\ &\qquad\qquad\qquad\qquad \to \langle\langle \{1,2\} \rangle\rangle \mathcal{F} \triangle_2 \{q\} \end{aligned}$$

Let $n = 2$, $R = (R_1, R_2)$ be a mechanism and $M = (s_1, s_2, \pi) \in \mathcal{M}_{fin}$, and let

$$\begin{aligned} R, M \models &\triangle_1\{p\} \wedge \triangle_2\{p \to q\} \\ &\wedge \langle\langle \emptyset \rangle\rangle \Box \left(\overset{\rightsquigarrow}{\triangle}_{12}\{\frac{t \sqcup \{a\}}{\{a\}}\} \wedge \overset{\rightsquigarrow}{\triangle}_{22}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{a, a \to b, b\}}\} \wedge \overset{\rightsquigarrow}{\triangle}_{22}\{\frac{t}{t}\} \right) \end{aligned}$$

Since $R, M \models \triangle_1\{p\} \wedge \triangle_2\{p \to q\}$, $p \in s_1$ and $p \to q \in s_2$. Note that

$$R, M \models \overset{\rightsquigarrow}{\triangle}_{12}\{\frac{t \sqcup \{a\}}{\{a\}}\} \wedge \overset{\rightsquigarrow}{\triangle}_{22}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{a, a \to b, b\}}\} \wedge \overset{\rightsquigarrow}{\triangle}_{22}\{\frac{t}{t}\}$$

Let $f_1 : \wp^{fin}(OL) \to (\wp^{fin}(OL))^2$ be such that for any s, $f_1(s) \in R_1(s)$, where

$$f_1(s_1) = (s_1^1, \{p\})$$

where s_1^1 is arbitrary. If $s \neq s_1$ then $f_1(s) \in R_1(s)$ is arbitrary ($R_1(s) \neq \emptyset$ for all s). Since $R, M \models \overset{\leadsto}{\triangle}_{12}\{\frac{t \sqcup \{a\}}{\{a\}}\}$, $[\![\frac{t \sqcup \{a\}}{\{a\}}]\!](s_1) = \{\{\alpha\} : \alpha \in s_1\} \subseteq R_1(s_1)^2$ (Lemma 7.3.1) and since $R, M \models \triangle_1\{p\}$, $\{p\} \in R_1(s_1)^2$. Thus, f_1 is well defined.

Let $f_2 : \wp^{fin}(OL) \to (\wp^{fin}(OL))^2$ be such that for any s, $f_2(s) \in R_2(s)$, where

$$f_2(s_2) = (s_1^2, s_2)$$
$$f_2(s_2 \cup \{p\}) = (s_1', s_2 \cup \{p\})$$

where s_1^2 and s_1' are arbitrary. If $s \neq s_2$ and $s \neq s_2 \cup \{p\}$ then $f_2(s) \in R_2(s)$ is arbitrary ($R_2(s) \neq \emptyset$ for all s). Since $R, M \models \overset{\leadsto}{\triangle}_{22}\{\frac{t}{t}\}$, $[\![\frac{t}{t}]\!](s_2) = \{s_2\} \subseteq R_j(s_2)^2$ (Lemma 7.3.1), so $f_2(s)$ is well defined when $s = s_2$.

Let λ be such that $\lambda[0] = (s_1, s_2)$ and $\lambda[1] = (s_1^1 \cup s_1^2, \{p\} \cup s_2)$. Since $(s_1^1, \{p\}) \in R_1(s_1)$ and $(s_1^2, s_2) \in R_2(s_2)$, λ is an R-computation. Since[7] $R, M \models \langle\langle\emptyset\rangle\rangle\Box \overset{\leadsto}{\triangle}_{22}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{a, a \to b, b\}}\}$, $R, (\lambda[1], \pi) \models \overset{\leadsto}{\triangle}_{22}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{a, a \to b, b\}}\}$. Thus, $[\![\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{a, a \to b, b\}}]\!](\{p\} \cup s_2) = \{\{p\} \cup s_2 \cup \{\beta\} : \alpha, \alpha \to \beta \in \{p\} \cup s_2\} \subseteq R_2(\{p\} \cup s_2)^2$ (see Example 7.1 on p. 88). Since $p \to q \in s_2$, $s_2 \cup \{p, q\} \in R_2(\{p\} \cup s_2)^2$, so $f_j(s)$ is also well defined when $s = s_2 \cup \{p\}$.

Since these functions are well defined, $\vec{f}_{\{1,2\}} \in Str(\{1,2\}, R)$ by definition. Let λ be as described above with $\lambda[2] = (f_1(\lambda[1]^1)^1 \cup f_2(\lambda[1]^2)^1, f_1(\lambda[1]^1)^2 \cup f_2(\lambda[1]^2)^2)$. Clearly, $\lambda \in out_R((s_1, s_2), \vec{f}_{\{1,2\}})$ and $q \in f_2(\lambda[1]^2)^2$, so

$$R, M \models \langle\langle\{1,2\}\rangle\rangle\mathcal{F} \triangle_2 \{q\} \qquad \Box$$

In Chapter 8, a more involved example is presented.

Monotonicity

Mechanisms are quite general; there are no restrictions on the possible new epistemic states. For example, a new state is not required to be an extension of the current state – an agent can forget. The formula

$$\overset{\leadsto}{\triangle}_{ii}\{\frac{t \sqcup \{a\}}{t}\}$$

describes an agent who may forget a single formula between the current and the next state. A common use of "forgetting" formulae is belief revision.

Often there is a need to model agents with *monotone* knowledge; i.e. agents with mechanisms which always extend the current state.

Definition 7.15 (Monotone Mechanism) If $R = (R_1, \ldots, R_n)$ is a mechanism, then R_i is *monotone* iff for all $s \in \wp^{fin}(OL)$:

$$s' \in R_i(s)^i \Rightarrow s \subseteq s'$$

If R_i is monotone, R is i-monotone. R is monotone iff it is i-monotone for each $i \in [1, n]$. $\Box$

[7]See a brief discussion on page 95.

We can axiomatize monotone mechanisms in the following way.

Lemma 7.5 Let R be a mechanism and $t, u \in V_T$.

$$R \models \overset{\leadsto}{\triangledown}_{ii}\{\frac{t}{t \sqcup u}\} \Leftrightarrow R \text{ is } i\text{-monotone} \tag{7.1}$$

□

PROOF By Lemma 7.3.2 the left side of (7.1) holds iff $R_i(s)^i \subseteq [\![T^R]\!](s)$ for every $s \in \wp^{fin}(OL)$, so it suffices to assume that $s' \in R_i(s)^i$ and show that $s \subseteq s' \Leftrightarrow s' \in [\![T^R]\!](s)$. $s' \in [\![T^R]\!](s) \Leftrightarrow \exists_{\Omega \in Subst} s = [t_\Omega]$ and $s' = [(t \sqcup u)_\Omega] = [t_\Omega] \cup [u_\Omega] \Leftrightarrow \exists_{T \in AL_T, U \in AL_T} s = [T]$ and $s' = [T] \cup [U]$ iff $s \subseteq s'$. ■

The consequence for computations is obvious:

Lemma 7.6 If R is i-monotone and λ is a computation in R, then $\lambda[k]^i \subseteq \lambda[k+1]^i$ for all $k \geq 0$. □

PROOF Follows directly from the definition of a computation (see p. 94) and of monotonicity. ■

Example 7.3 Another version of the formula from Example 7.2 (p. 99) is:

$$\begin{aligned}&(\triangle_1\{p\} \wedge \triangle_2\{p \to q\} \\ &\quad \wedge \langle\langle\emptyset\rangle\rangle\Box \left(\overset{\leadsto}{\triangle}_{12}\{\frac{t \sqcup \{a\}}{\{a\}}\} \wedge \overset{\leadsto}{\triangle}_{22}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{b\}}\} \wedge \overset{\leadsto}{\triangledown}_{22}\{\frac{t}{t \sqcup u}\}\right)) \\ &\qquad\qquad\qquad\qquad \to \langle\langle\{1,2\}\rangle\rangle\mathcal{F}\triangle_2\{q\}\end{aligned}$$

The difference is that in the first version, it is required that agent 2 *can* do monotone reasoning – more precisely either monotone modus ponens or recall of current state without new inferences. In the current version, it is required that agent 2 *must* do monotone reasoning. Thus, the modus ponens rule in the current version need not specify explicitly that the two arguments in the antecedent must be remembered – but they will always be remembered also in this example.

It is easy to see that also the current version is valid. □

Rules which imply monotonicity are called monotone rules:

Definition 7.16 (Monotone Rules) A rule term $T^R \in TRL_T$ is monotone iff

$$\models \overset{\leadsto}{\triangledown}_{ii}T^R \to \overset{\leadsto}{\triangledown}_{ii}\{\frac{t}{t \sqcup u}\}$$

□

The following lemma follows directly.

Lemma 7.7 If $R \models \overset{\leadsto}{\triangledown}_{ii}T^R$ and T^R is monotone, then R is i-monotone. □

In particular, if $T_1^R, \ldots, T_n^R$ are complete and monotone, there exists an i-monotone mechanism R such that $R \models \wedge_{j \in [1,n]}\overset{\leadsto}{\Diamond}_{ij}T_j^R$.

A common pattern of specifying monotone rules is

$$\frac{t \sqcup T}{t \sqcup T \sqcup S}$$

for some T, S, illustrated e.g. by the modus ponens rule in Example 7.2 on p. 99. The following is defined as a shorthand notation for this pattern:

$$\frac{T}{\underline{S}} \equiv \frac{t \sqcup T}{t \sqcup T \sqcup S}$$

The type of monotonicity discussed above is monotonicity of *knowledge*. It is also possible to talk about monotonicity of *rules*. Consider an agent with monotone knowledge. If the agent can use a rule R in state s, it is not necessarily the case that he can use the same rule in the same way in a state $s' \supseteq s$ even if he knows the rule in both states. The reason is that the antecedents of rules match *complete* epistemic states. Typically, we want to use *rule-monotone rules* to model mechanisms — rules which can be used in the same way in an extension of a state. A possible definition of rule-monotonicity of a rule term $T^R \in TRL_T$, is that each rule $\frac{T^A}{T^C}$ is on the form

$$\frac{t \sqcup T}{(t \sqcap V) \sqcup U}$$

where t does not occur in T, U or V. According to this definition, a monotone rule can be used in the "same" way in an extended state – the only additional formulae in the consequent is formulae already in the extended state. It is easy to see that if T^R is a rule-monotone term, according to the mentioned definition, then $s \subseteq r$ and $s' \in [\![T^R]\!](s)$ implies that there is a $r' \in [\![T^R]\!](r)$ such that $s' \subseteq r'$. Monotonicity of rules will not be discussed further here.

7.6.2 Rules and Temporal Properties

Like rules, formulae with temporal connectives can express facts about the mechanism.

A *TEL* formula is a statement about a particular state; the contents of the state (expressed with *EL* formulae) and/or how that state relates to other states (expressed with rule- or temporal connectives). For example, we use $\triangle_1\{p\}$ to say that agent 1 knows p *now*. We can use $\langle\langle 1\rangle\rangle\Box \triangle_1 \{p\}$ to say that agent 1 *can* ensure that he will always know p. If we want to say that agent 1 *always* will know p, we must use $\langle\langle\emptyset\rangle\rangle\Box \triangle_1 \{p\}$. The same is the case for rules. $\overset{\leadsto}{\triangle}_{ij}T^R$ only means that agent i knows the rules T^R *now*. Although an agent's mechanism is fixed, an agent may know different rules in different states. For example, there are rules T^R such that

$$\not\models \overset{\leadsto}{\triangle}_{ij}T^R \rightarrow \langle\langle\{i\}\rangle\rangle \bigcirc \overset{\leadsto}{\triangle}_{ij}T^R$$

If we want to say that an agent will know a rule always in the future, we must use an expression like the one for knowledge: $\langle\langle\emptyset\rangle\rangle\Box \overset{\leadsto}{\triangle}_{ij} T^R$. The formula in Example 7.2 (p. 99) expresses that the agents know certain formulae in the current state, and that they know certain rules and will remember them in the future.

There is, however, a circumstance in which an agent implicitly will know in a future state a rule it knows in the current state: if the future epistemic state is equal to the current epistemic state for the agent. From the agent's point of view, the world is the same in the two states – he cannot discern between the current state and the future state. Being at the same point in the lattice, by the definition of a mechanism (as relation on epistemic states), he must thus be able to do exactly the same actions. Formally, let $1 \leq i \leq n$:

Lemma 7.8

$$\models \Diamond_i T \to \big(\bigwedge_{1 \leq j \leq n} \overset{\rightsquigarrow}{\Diamond}_{ij} T_j^R \leftrightarrow \langle\langle \emptyset \rangle\rangle \Box (\Diamond_i T \to \bigwedge_{1 \leq j \leq n} \overset{\rightsquigarrow}{\Diamond}_{ij} T_j^R)\big) \qquad \Box$$

PROOF Let $R = (R_1, \ldots, R_n)$ and $M = (s_1, \ldots, s_n, \pi)$. If $R, M \models \Diamond_i T \wedge \wedge_{1 \leq j \leq n} \overset{\rightsquigarrow}{\Diamond}_{ij} T_j^R$ then $s_i = [T]$ and $R_i([T]) = [\![T_1^R]\!]([T]) \times \cdots \times [\![T_n^R]\!]([T])$ (Lemma 7.3.4). If λ is a computation and $R, (\lambda[k], \pi) \models \Diamond_i T$, then also $R, (\lambda[k], \pi)) \models \wedge_{1 \leq j \leq n} \overset{\rightsquigarrow}{\Diamond}_{ij} T_j^R$. The other direction (to the left) of the double implication is (also) trivial. ■

Lemma 7.8 is a syntactic statement about incomplete information, as defined semantically in Def. 7.9 on p. 91 (see also the introduction in Section 7.4). It is discussed further in Section 9.4.

The rule formula $\overset{\rightsquigarrow}{\triangle}_{ij} T^R$ is a statement about agent i's capability to enforce certain properties of the next state of the system — an informal description also often stated about an ATL formula such as $\langle\langle \{i\} \rangle\rangle \bigcirc \phi$. For example, $\overset{\rightsquigarrow}{\triangle}_{ij} \{\frac{p}{q}\}$ and $\triangle_i \{p\} \to \langle\langle \{i\} \rangle\rangle \bigcirc \triangle_j \{q\}$ may seem to express the same thing. However, they are not equivalent. One direction holds:

$$\models \overset{\rightsquigarrow}{\triangle}_{ij} \{\frac{p}{q}\} \to (\triangle_i \{p\} \to \langle\langle \{i\} \rangle\rangle \bigcirc \triangle_j \{q\})$$

but not the other:

$$\not\models \overset{\rightsquigarrow}{\triangle}_{ij} \{\frac{p}{q}\} \leftarrow (\triangle_i \{p\} \to \langle\langle \{i\} \rangle\rangle \bigcirc \triangle_j \{q\})$$

The reason for this is that $\overset{\rightsquigarrow}{\triangle}_{ij} \{\frac{p}{q}\}$ is indeed a statement about i's mechanism, while $\langle\langle \{i\} \rangle\rangle \bigcirc \triangle_j \{q\}$ is not necessarily. For example, it may be the case that

$$R, M \models \langle\langle \emptyset \rangle\rangle \bigcirc \triangle_j \{q\}$$

which implies that $R, M \models \langle\langle \{i\} \rangle\rangle \bigcirc \triangle_j \{q\}$, because some other agent's mechanism always sends q to j. In other words, rule connectives express something about an agent's mechanism and therefore about possible future states, while temporal connectives express something about future states without regards to how these states come about. This point is discussed further in a comparison with ATEL in Section 9.4. The following holds:

$$\models (\triangle_i \{p\} \to (\langle\langle \{i\} \rangle\rangle \bigcirc \triangle_j \{q\} \wedge \neg \langle\langle \emptyset \rangle\rangle \bigcirc \triangle_j \{q\})) \to \overset{\rightsquigarrow}{\triangle}_{ij} \{\frac{p}{q}\}$$

but now the opposite direction does not hold:

$$\not\models (\triangle_i \{p\} \to (\langle\langle \{i\} \rangle\rangle \bigcirc \triangle_j \{q\} \wedge \neg \langle\langle \emptyset \rangle\rangle \bigcirc \triangle_j \{q\})) \leftarrow \overset{\rightsquigarrow}{\triangle}_{ij} \{\frac{p}{q}\}$$

Another important way in which rules are more general than temporal formulae is that due to the use of variables, an infinite number of temporal formulae may be needed to express the equivalent of a single rule. For example, monotonicity can be expressed with the following schema:

$$\triangle_i T \rightarrow \langle\langle\emptyset\rangle\rangle \Box \triangle_i T$$

Of course, rule- and temporal formulae may together be inconsistent. For example,

$$\models \neg(\triangle_i\{p\} \wedge \overset{\rightsquigarrow}{\bigtriangledown}_{ii}\{\frac{t}{t \sqcup u}\} \wedge \langle\langle\{i\}\rangle\rangle \bigcirc \neg \triangle_i \{p\})$$

Communication

Semantically, communication is defined via the definition of the new epistemic state of an agent as the union of all the agents' choices for that agent. Here, we look at some syntactic expressions of the properties of communication.

One property of communication is that an agent cannot force another agent to *forget* something, since he can only *add* to the next epistemic state of the other agent. Of course, that agent i cannot force j to forget p only makes sense if j can remember p in the first place. This property can be expressed as

$$\langle\langle\{j\}\rangle\rangle \bigcirc \triangle_j\{p\} \rightarrow [\![\{i\}]\!] \bigcirc \triangle_j\{p\}$$

(the dual path quantifier $[\![A]\!]$ is defined in Section 2.3.2). Forgetting means here not only not remembering the formulae from the current epistemic state, but also not remembering communication from other agents between the current and the next state. The property above is an instance of the following Lemma 7.9.1.

Lemma 7.9

1. When $i \notin \Gamma$:
$$\models \langle\langle\Gamma\rangle\rangle \bigcirc \triangle_j T \rightarrow [\![\{i\}]\!] \bigcirc \triangle_j T$$

2. When $\Gamma \cap \Gamma' = \emptyset$:
$$(\langle\langle\Gamma\rangle\rangle \bigcirc \triangle_j T \wedge \langle\langle\Gamma'\rangle\rangle \bigcirc \triangle_j T') \rightarrow \langle\langle\Gamma \cup \Gamma'\rangle\rangle \bigcirc \triangle_j (T \sqcup T') \qquad \Box$$

PROOF

1. If $R, (s_1, \ldots, s_n, \pi) \models \langle\langle\Gamma\rangle\rangle \bigcirc \triangle_j T$, then for each $\alpha \in [T]$ there is a $k_\alpha \in \Gamma$ such that $\alpha \in d_{k_\alpha}(s_{k_\alpha})$. If $R, (s_1, \ldots, s_n, \pi) \models \langle\langle\{i\}\rangle\rangle \bigcirc \neg \triangle_j T$, there is a strategy f_i for i such that the next epistemic state of j does not include $[T]$ no matter what the other agents do. But each k_α can choose $d_{k_\alpha}(s_{k_\alpha})$, so $R, (s_1, \ldots, s_n, \pi) \models \neg\langle\langle\{i\}\rangle\rangle \bigcirc \neg \triangle_j T$.

2. Obvious: agents in Γ can use the strategy they can use to enforce that j knows at least $[T]$ and agents in Γ' can use the strategy they can use to enforce that j knows at least $[T']$. ■

7.7 No Communication

In the special case of agents who only reasons but do not communicate, the framework can be simplified.

A mechanism $R = (R_1, \dots, R_n)$ is *without communication* iff

$$R_i(s)^j = \{\emptyset\}$$

for all $i \neq j, s$.

Since R is completely determined by $R_i(s)^i$, for all i and s, we can abuse notation and write $s_i \in R_i(s)$ for a tuple $(\emptyset, \dots, s_i, \dots, \emptyset) \in R_i(s)$. Similarly for strategies; we write $f_i(s) = s_1$ as a shorthand for $f_i(s) = (\emptyset, \dots, s_i, \dots, \emptyset)$. The transition function is also simplified; $\delta(q, (s_1, \dots, s_n)) = (s_1, \dots, s_n)$.

The satisfaction relation restricted to mechanisms with no communication is denoted $\models_{nc}$.

Clearly, if R is a mechanism without communication, then the induced concurrent game structure is Moore synchronous (see sec. 2.3.3)[8].

Mechanisms with no communication are illustrated in the example in the next section.

7.7.1 Example: Three Wise Men

The *Three Wise Men* puzzle is a well known example of reasoning about knowledge in multi-agent systems.[9]

In this Section, a selected aspect of this puzzle is modeled in order to demonstrate reasoning with rules in DSEL. First, the problem is reviewed.

Example 7.4 (Three Wise Men) A certain king wishes to test his three wise men. He arranges them in a circle so that they can see and hear each other and tells them that he will put a white or black spot on each of their foreheads but that at least one spot will be white. In fact all three spots are white. He then repeatedly asks them, "Do you know the color of your spot?" What do they answer? □

SOLUTION All the three wise men answer "no" the first two times, and "yes" the third. Each wise man reasons as follows. Right after the first question, he perceives the color of the spots on the two other men, but this is not enough to deduce the color of his own spot so he answers "no". Then he hears the answers from the two other wise men. When the second man answers "no", the first man knows that at least one of his own and the third man's spot must be white or else the second man would have answered "yes" since it is common knowledge that at least one spot is white. Similarly, from the third man's answer, the first man gains the information that his own or the second man's spot (or both) must be white. This is still not enough information to deduce his

[8]In addition, the intention of the first condition on turn-based structures with incomplete information, i.e. that a player can only affect his observable propositions, holds, as mentioned in the introduction to Sec. 7.4.

[9]This problem belongs to a class of similar puzzles from the folklore, including *The Muddy Children Puzzle*, *The Cheating Wives Puzzle* and *Conway's Paradox*, among others. The version given here is from McCarthy (1978), one of the first appearances of the problem in the computer science literature.

own color, so he answers “no” to the second question too. The first man knows that the second man reasons exactly as he does, thus he now knows that the second man knows that either the first or the second man (or both) has a white spot. When the second man answers “no” to the second question, the first man knows that his own color must be white since otherwise the second man would have answered “yes”. The first wise man therefore answers “yes” to the third question. ■

The selected, in order to focus on reasoning rather than communication, aspect of the solution modeled below is the reasoning of the first wise man immediately after the second question is answered by all the wise men.

Axiomatization

The situation is modeled by three agents representing the wise men, and three primitive propositions p_1, p_2, p_3 where p_i means that agent i has a white spot.

In Figure 7.3 a rule term is presented. It includes rules for doing propositional reasoning, and for reasoning about other agents' reasoning. These rules have been selected in order to demonstrate meaningful reasoning in this small example.

In addition to giving agents rules, initial knowledge and knowledge obtained through observation must be defined. This includes facts such as

$$\neg p_1 \rightarrow \triangle_2\{\neg p_1\}$$

(agent 2 can observe agent 1), but also nested knowledge such as

$$\triangle_3\{\neg p_1 \rightarrow \triangle_2\{\neg p_1\}\}$$

(agent 3 knows the previous fact). Usually, this type of knowledge is modeled as common knowledge. Since the concept of common knowledge entails partial logical omniscience, nested knowledge is here modeled explicitly.

The situation being modeled here is the situation immediately after the second question is answered. Then, agent 1 knows that

- agent 2 does not know p_2, from observing agent 2's answer to the second question
- agent 2 knows that 3 does not know p_3, from observing that agent 2 observed agent 3's answer to the first question

These observations are collected in Figure 7.4, which presents a term T representing the initial knowledge of an agent (for brevity, only observations needed in the following proof is included).

It is assumed that the agents do not communicate in this part of the puzzle, therefore they will be modeled by mechanisms without communication as presented above.

A Result

I show that with the given rules and initial knowledge, agent 1 can deduce the colour of his own spot, i.e. that:

$$\models_{nc} (\Diamond_1 T \wedge \overset{\rightsquigarrow}{\Diamond}_{11} T^R) \rightarrow \langle\langle 1 \rangle\rangle \mathcal{F} \triangle_1 \{p_1\} \qquad (7.2)$$

$$T^R = \left\{ \frac{\{a, a \to b\}}{\underline{\{b\}}} \right\} \quad \textbf{(MP)}$$

$$\sqcup \left\{ \frac{\{\triangle_j\{a \to b\}\}}{\underline{\{\triangle_j\{a\} \to \triangle_j\{b\}\}}} : j \in [1,3] \right\} \quad \textbf{(Distr2)}$$

$$\sqcup \left\{ \frac{\{\triangle_j\{\triangle_k\{a \to (b \to c)\}\}\}}{\underline{\triangle_j\{\triangle_k\{a\} \to (\triangle_k\{b\} \to \triangle_k\{c\})\}\}}} : k, j \in [1,3] \right\} \quad \textbf{(Distr3)}$$

$$\sqcup \left\{ \frac{\{a \to b, b \to c\}}{\underline{\{a \to c\}}} \right\} \quad \textbf{(Trans)}$$

$$\sqcup \left\{ \frac{\{\triangle_j\{a \to b\}, \triangle_j\{b \to c\}\}}{\underline{\{\triangle_j\{a \to c\}\}}} : j \in [1,3] \right\} \quad \textbf{(Trans2)}$$

$$\sqcup \left\{ \frac{\{\neg a \to b\}}{\underline{\{\neg b \to a\}}} \right\} \quad \textbf{(Prop1)}$$

$$\sqcup \left\{ \frac{\{\triangle_j\{a \to (b \to c)\}\}}{\underline{\{\triangle_j\{b \to (a \to c)\}\}}} : j \in [1,3] \right\} \quad \textbf{(Prop2)}$$

$$\sqcup \left\{ \frac{\{\triangle_j\{c \to (\neg b \to a)\}\}}{\underline{\{\triangle_j\{c \to (\neg a \to b)\}\}}} : j \in [1,3] \right\} \quad \textbf{(Prop3)}$$

$$\sqcup \left\{ \frac{\{\triangle_j\{\triangle_k\{a \vee b \vee c\}\}\}}{\underline{\{\triangle_j\{\triangle_k\{\neg a \to (\neg b \to c)\}\}\}}} : k, j \in [1,3] \right\} \quad \textbf{(Prop4)}$$

Figure 7.3: The rule term T^R. Recall the underlined notation, as a shorthand for monotone rules, introduced on p. 102.

$$\begin{aligned}
T = &\{\neg p_h \to \triangle_g\{\neg p_h\} : g, h \in [1,3], g \neq h\} && \textbf{(Obs1)}\\
&\sqcup\{\triangle_j\{\neg p_h \to \triangle_g\{\neg p_h\}\} : j, g, h \in [1,3], g \neq h\} && \textbf{(Obs2)}\\
&\sqcup\{\triangle_j\{\triangle_k\{p_1 \vee p_2 \vee p_3\}\} : k, j \in [1,3]\} && \textbf{(Obs3)}\\
&\sqcup\{\neg \triangle_j \{p_j\} : j \in [1,3]\} && \textbf{(Obs4)}\\
&\sqcup\{\triangle_j\{\neg \triangle_k \{p_k\}\} : k, j \in [1,3]\} && \textbf{(Obs5)}
\end{aligned}$$

Figure 7.4: Initial knowledge: the term T

Let $R, (\vec{s}, \pi) \models_{nc} (\Diamond_1 T \wedge \overset{\rightsquigarrow}{\Diamond}_{11} T^R)$. Let $f_1 : \wp^{fin}(OL) \to \wp^{fin}(OL)$ be such that $f_1(s) \in R_1(s)$ for all s and $f_1(x_i) = x_{i+1}$ for $i \in [0, 13]$ where x_i is defined below (explanation follows):

$$\begin{array}{rlll}
x_0 &= [T] & & (1)\\
&= x_0 \cup \{\triangle_2\{\triangle_3\{p_1 \vee p_2 \vee p_3\}\}\} & \textbf{Obs3} & (2)\\
x_1 &= x_0 \cup \{\triangle_2\{\triangle_3\{\neg p_1 \to (\neg p_2 \to p_3)\}\}\} & \textbf{Prop4}, (2) & (3)\\
x_2 &= x_1 \cup \{\triangle_2\{\triangle_3\{\neg p_1\} \to (\triangle_3\{\neg p_2\} \to \triangle_3\{p_3\})\}\} & \textbf{Distr3}, (3) & (4)\\
&= x_2 \cup \{\triangle_2\{\neg p_1 \to \triangle_3\{\neg p_1\}\}\} & \textbf{Obs2} & (5)\\
x_3 &= x_2 \cup \{\triangle_2\{\neg p_1 \to (\triangle_3\{\neg p_2\} \to \triangle_3\{p_3\})\}\} & \textbf{Trans2}, (4), (5) & (6)\\
x_4 &= x_3 \cup \{\triangle_2\{\triangle_3\{\neg p_2\} \to (\neg p_1 \to \triangle_3\{p_3\})\}\} & \textbf{Prop2}, (6) & (7)\\
&= x_4 \cup \{\triangle_2\{\neg p_2 \to \triangle_3\{\neg p_2\}\}\} & \textbf{Obs2} & (8)\\
x_5 &= x_4 \cup \{\triangle_2\{\neg p_2 \to (\neg p_1 \to \triangle_3\{p_3\})\}\} & \textbf{Trans2}, (7), (8) & (9)\\
x_6 &= x_5 \cup \{\triangle_2\{\neg p_1 \to (\neg p_2 \to \triangle_3\{p_3\})\}\} & \textbf{Prop2}, (9) & (10)\\
x_7 &= x_6 \cup \{\triangle_2\{\neg p_1 \to (\neg \triangle_3 \{p_3\} \to p_2)\}\} & \textbf{Prop3}, (10) & (11)\\
x_8 &= x_7 \cup \{\triangle_2\{\neg \triangle_3 \{p_3\} \to (\neg p_1 \to p_2)\}\} & \textbf{Prop2}, (11) & (12)\\
x_9 &= x_8 \cup \{\triangle_2\{\neg \triangle_3 \{p_3\}\} \to \triangle_2\{\neg p_1 \to p_2\}\} & \textbf{Distr2}, (12) & (13)\\
&= x_9 \cup \{\triangle_2\{\neg \triangle_3 \{p_3\}\}\} & \textbf{Obs5} & (14)\\
x_{10} &= x_9 \cup \{\triangle_2\{\neg p_1 \to p_2\}\} & \textbf{MP}, (13), (14) & (15)\\
x_{11} &= x_{10} \cup \{\triangle_2\{\neg p_1\} \to \triangle_2\{p_2\}\} & \textbf{Distr2}, (15) & (16)\\
&= x_{11} \cup \{\neg p_1 \to \triangle_2\{\neg p_1\}\} & \textbf{Obs1} & (17)\\
x_{12} &= x_{11} \cup \{\neg p_1 \to \triangle_2\{p_2\}\} & \textbf{Trans}, (16), (17) & (18)\\
x_{13} &= x_{12} \cup \{\neg \triangle_2 \{p_2\} \to p_1\} & \textbf{Prop1}, (18) & (19)\\
&= x_{13} \cup \{\neg \triangle_2 \{p_2\}\} & \textbf{Obs4} & (20)\\
x_{14} &= x_{13} \cup \{p_1\} & \textbf{MP}(19, 20) & (21)
\end{array}$$

This list defines the sets $x_0, \ldots, x_{14}$ through which agent 1 will step consecutively in his reasoning. Some sets are listed more than once in order to present several formulas within the set, with an explanation in the middle column. It is easy to see that these sets are all different, so that f_1 is well defined. It is easy to see, by the explanation in the middle column, that $f(x_i) \in R_1(x_i)$ for $i \in [0, 13]$, since $R_1(x_i)^1 = [\![T^R]\!](x_i)$. Thus, $f_1 \in Str(\{1\}, R)$.

Let $\lambda \in out_R(\{f_1\}, \vec{s})$. It is easy to see that since agent 1 uses the strategy f_1 and since $\lambda[0]^1 = s_1 = [T] = x_0$, $\lambda[k]^1 = x_k$ for $k \in [0, 14]$ and thus that

$$R, (\lambda[14], \pi) \models_{nc} \triangle_1\{p_1\}$$

Thus, (7.2) holds.

Chapter 8

Example: The Byzantine Generals Problem

8.1 Introduction

The *Byzantine Generals Problem* (Lamport, Shostak, & Pease, 1982) is used to discuss properties of systems of communicating processors, some of which may be faulty.

The problem is as follows. A commanding general and a group of $n - 1$ other generals are camped outside a city, and can only communicate via direct and reliable messages. Some of the generals are *traitors* and the others are called *loyal* generals, but no one knows who the traitors are or who the loyal generals are. The commanding general sends a message to the other generals which is either to attack or not to attack. The loyal generals *must* agree, possibly after some communication, on a common plan: to attack or not to attack. Thus, if the commanding general is loyal, all other loyal generals must obey the message he sends. The traitors may try to confuse the other generals in order to make them fail in finding a common plan. The problem is: is it possible to equip the loyal generals with algorithms (or "protocols") which ensure that they will agree on a common plan no matter what the traitors do?

It is well known that the answer is "yes" if and only if more than two thirds of the generals are loyal. The purpose of the case study in this chapter is to illustrate some aspects of the logic DSEL defined in Chapter 7 such as expressiveness; the results which are shown are well known and the proofs of the results are partly adaptions of proofs from the literature to the DSEL framework. "Partly", because the DSEL framework requires a very detailed modeling of reasoning and communication and several assumptions made implicit in other higher level proofs must be made explicit in DSEL.

8.2 Axiomatization

"Agreeing on a common plan" by communication is called reaching *interactive consistency*. Interactive consistency for a group G of loyal generals, $IC(G)$, is defined as follows.

Let the generals be named $Ags = \{1, \ldots, n\}$ and let $cg \in Ags$ be the commanding general.

Definition 8.1 ($IC(G, \alpha), IC(G)$) Given a group of agents G and a formula $\alpha \in AL$, the formula $IC(G, \alpha)$ is the conjunction of the following formulae:

1. $\bigwedge_{i \in G}((\triangle_i\{\alpha\} \vee \triangle_i\{\neg\alpha\}) \wedge \neg(\triangle_i\{\alpha\} \wedge \triangle_i\{\neg\alpha\}))$
2. $\bigwedge_{i,j \in G}(\triangle_i\{\alpha\} \leftrightarrow \triangle_j\{\alpha\})$

$IC(G, \triangle_{cg}\{attack\})$ is abbreviated $IC(G)$. □

$IC(G)$ means that the loyal generals G have interactive consistency about the commanding general's decision. Here, having decided is modeled as having deduced a formula. Knowing the formula $\triangle_{cg}\{attack\}$ means knowing that the commanding general has decided to attack, knowing $\neg \triangle_{cg} \{attack\}$ means knowing that he has not decided to attack. The first part of $IC(G)$ ensures that all loyal generals have consistently decided whether to attack or not, the second part that they have made the same decision. When the commanding general is loyal, interactive consistency implies that every loyal general will obey him.

In the following definition Γ describes the starting conditions. They include the fact that the commanding general knows what he has decided, so that we can defined interactive consistency as shared knowledge of the formula $\triangle_{cg}\{attack\}$ or $\neg \triangle_{cg} \{attack\}$. In addition, a few more assumptions are made regarding the commanding general's initial state. These are made in order to simplify the proof; they are not strictly necessary and could be modeled with reasoning steps, but the focus in this example is on communication.

Definition 8.2 (Γ) Let Γ be the conjunction of the following formulae:

1. $\Diamond_{cg}\{\triangle_{cg}\{attack\}, \triangle_{cg}\{\neg\neg attack\}\} \vee \Diamond_{cg}\{\neg \triangle_{cg} \{attack\}, \triangle_{cg}\{\neg attack\}\}$
2. $\bigwedge_{i \neq cg} \bigtriangledown_i \emptyset$ □

Γ states that the commanding general has decided, and that the other generals (loyal or not) have empty epistemic states. The latter is required, since we want to model a starting situation in which the generals have not exchanged any information yet.

A formal description of the problem has many aspects, and one of them is the type of communication allowed — i.e. which kinds of messages the agents can send. First, a formulation without any restrictions on communication; such restrictions will be discussed shortly.

The problem can be formulated as follows: does there exist complete, monotone and deterministic rules T^{ij} for all $i, j \in [1, n]$ such that for certain groups of agents G,

$$\models \left(\langle\langle \emptyset \rangle\rangle \Box \bigwedge_{i \in G, j \in Ags} \overset{\leadsto}{\Diamond}_{ij} T^{ij} \right) \rightarrow (\Gamma \rightarrow \langle\langle G \rangle\rangle \mathcal{F} IC(G)) \tag{8.1}$$

holds? In this formulation, G plays the role of the loyal agents, and there are less than or equal to $n - |G|$ traitors. The formula must be valid for *all* of the "certain groups" G. Informally: does there exist rules such that for any G (in the "certain groups"), if the rules completely describe the mechanisms of the agents in G, G can cooperate to achieve interactive consistency from the situation described by Γ no matter what the other agents do? The rules are required to be:

- Complete: the rules should describe what the agents should do in *any* situation.
- Monotone: we assume that agents remember; a not unreasonable assumption about the power of the reasoning mechanisms.
- Deterministic: the rules *must* be deterministic in order to illustrate the problem faithfully. The formula (8.1) is universally quantified over the "certain groups"; it is required to hold for the *same* rules T^{ij} for *different* G. This is crucial, because the generals should not be able to take different actions based upon how many traitors there are or other facts which they do not know — they should act based only on their epistemic states. The problem is whether there is a protocol which describes what they should do in each epistemic state, and such a protocol is modeled by (complete and) deterministic rules[1].

However, considering the nature of communication in DSEL the answer to the problem as stated above is clearly "no", since traitors can potentially "insert" any formula into the epistemic states of the generals. Thus, a restriction on the messages is needed. I model this restriction as *oral messages*.

8.2.1 Oral Messages

The concept of oral messages is as follows. First, every message that is sent must be delivered correctly. This can be modelled by letting a message be a formula α, and an agent i sends a message to j by choosing an action $(s_1, \ldots, s_n)$ where $s_j = \{\alpha\}$. Second, the receiver of a message should know who sent it. This can be modeled by restricting the communication part of the mechanisms for all agents. Formally, let $Oral(i,j)$ be the following rule term (explanation follows):

$$Oral(i,j) = \{\frac{t}{\triangle_j\{\neg\neg\triangle_i\{a\}\}}, \frac{t}{\triangle_j\{\neg\triangle_i\{a\}\}}, \frac{t}{\emptyset}\}$$

The following is required to hold:

$$\bigwedge_{i,j} \overset{\leadsto}{\triangledown}_{ij} Oral(i,j)$$

This restriction on messages may seem confusing, but it is just a way of encoding the fact that "this is something I received from i" in j's epistemic state, in the available language. The use of the double negation is a technical device to support this encoding. An agent i can only send messages which are formulae

[1]A consequence of determinism is that the notion of strategies for the agents G becomes unnecessary. In fact the operator $\langle\langle G\rangle\rangle$ in eq. (8.1) can be written as $\langle\langle\emptyset\rangle\rangle$ since G have deterministic mechanisms.

on the form $\triangle_j\{\neg\neg \triangle_i \{\alpha\}\}$ or $\triangle_j\{\neg \triangle_i \{\alpha\}\}$, where α is a formula (not necessarily known by i) to agent j — or send nothing at all. Agent j can then interpret the formulae as messages from i. The encoding ensures that messages will not be confused with inferences in the model that will be constructed shortly. This also allows the third assumption on oral messages to hold: the absence of a message can be detected.

A rule term $T^R \in TRL_T$ is *oral* iff

$$\models \overset{\leadsto}{\bigtriangledown}_{ij} T^R \rightarrow \overset{\leadsto}{\bigtriangledown}_{ij} Oral(i,j) \tag{8.2}$$

Thus, the formulation of the problem with restrictions on the messages is obtained by requiring that the T^{ij} be oral (for all $i \neq j$) in addition to being complete, deterministic and monotone, and by also restricting the traitors to oral messages:

$$\models \langle\langle\emptyset\rangle\rangle\Box \left(\bigwedge_{i\in G, j\in Ags} \overset{\leadsto}{\Diamond}_{ij} T^{ij} \wedge \bigwedge_{i\in(Ags\setminus G), j\in Ags} \overset{\leadsto}{\bigtriangledown}_{ij} Oral(i,j) \right) \rightarrow (\Gamma \rightarrow \langle\langle G\rangle\rangle \mathcal{F}IC(G))$$

In the next section, I show that the formula does not hold for all $|G| \geq \lfloor \frac{2n}{3} \rfloor$, while in Section 8.4 a related positive result is shown.

8.3 An Impossibility Result

The following is a statement of the impossibility result: it is not possible to equip agents with rules so that even if at least two thirds of the agents are loyal they are always guaranteed to be able to obtain interactive consistency. The proof is an adaption of a proof by Pease, Shostak, & Lamport (1980).

Theorem 8.1 There does not exist a combination of complete and deterministic rules T^{ij} for every agent i and j such that T^{ii} is monotone for each i, T^{ij} is oral for each $i \neq j$ and for every $G \subseteq Ags$ such that $|G| \geq \lfloor \frac{2n}{3} \rfloor$

$$\models \langle\langle\emptyset\rangle\rangle\Box \left(\bigwedge_{i\in G, j\in Ags} \overset{\leadsto}{\Diamond}_{ij} T^{ij} \wedge \bigwedge_{i\in(Ags\setminus G), j\in Ags} \overset{\leadsto}{\bigtriangledown}_{ij} Oral(i,j) \right) \rightarrow (\Gamma \rightarrow \langle\langle G\rangle\rangle \mathcal{F}IC(G)) \tag{8.3}$$

$\Box$

PROOF Assume the opposite, i.e. that there exist complete and deterministic $\{T^{ij} : 1 \leq i \leq n, 1 \leq j \leq n\}$ such that T^{ii} is monotone for each i, T^{ij} oral for $i \neq j$ and (8.3) holds for every $|G| \geq \lfloor \frac{2n}{3} \rfloor$.

Divide the n agents into three groups A, B and C such that $A \cup B \cup C = Ags$, $cg \in C$ and $|X| \leq \lceil \frac{n}{3} \rceil$ for each $X \in \{A, B, C\}$. It is easy to see[2] that:

$$|A \cup C| \geq \left\lfloor \frac{2n}{3} \right\rfloor \qquad |A \cup B| \geq \left\lfloor \frac{2n}{3} \right\rfloor \qquad |B \cup C| \geq \left\lfloor \frac{2n}{3} \right\rfloor$$

I now construct a mechanism R^{AC} which, informally speaking, is the least restrictive mechanism (maximal model) in which the mechanisms of each agent

[2]Since $n - \lceil \frac{n}{3} \rceil = \lfloor \frac{2n}{3} \rfloor$.

$i \in A \cup C$ is described by T^{ij} (for each $j \in Ags$) and the mechanisms of the agents in B are only restricted to oral messages. Formally, let $R^{AC} = (R_1^{AC}, \dots, R_n^{AC})$ where

$$R_i^{AC}(s) = \begin{cases} [\![T^{i1}]\!](s) \times \cdots \times [\![T^{1n}]\!](s) & i \in A \cup C \\ [\![Oral(i,1)]\!](s) \times \cdots \times [\![Oral(i,n)]\!](s) & \text{otherwise} \end{cases}$$

R_i^{AC} is a mechanism since each T^{ij} is complete and thus $[\![T^{ij}]\!](s) \neq \emptyset$.

Similarly, I construct mechanisms R^{AB} and R^{BC} in which $A \cup B$ are determined by T^{ij} and C are only restricted to oral messages, and $B \cup C$ determined by T^{ij} and A only restricted to oral messages, respectively. Let $R^{AB} = (R_1^{AB}, \dots, R_n^{AB})$ where

$$R_i^{AB}(s) = \begin{cases} [\![T^{i1}]\!](s) \times \cdots \times [\![T^{1n}]\!](s) & i \in A \cup B \\ [\![Oral(i,1)]\!](s) \times \cdots \times [\![Oral(i,n)]\!](s) & \text{otherwise} \end{cases}$$

Let $R^{BC} = (R_1^{BC}, \dots, R_n^{BC})$ where

$$R_i^{BC}(s) = \begin{cases} [\![T^{i1}]\!](s) \times \cdots \times [\![T^{1n}]\!](s) & i \in B \cup C \\ [\![Oral(i,1)]\!](s) \times \cdots \times [\![Oral(i,n)]\!](s) & \text{otherwise} \end{cases}$$

Let $i \in A \cup C$ and $1 \leq j \leq n$, and let $\vec{s} = (s_1, \dots, s_n)$ and π be arbitrary. By Lemma 7.3.5

$$R^{AC}, (\vec{s}, \pi) \models \overset{\rightsquigarrow}{\Diamond}_{ij} T^{ij} \text{ } (\vec{s}, \pi, j \text{ arbitrary}, i \in A \cup C) \tag{8.4}$$

Since, for each $i \in A \cup C$, $R^{AC} \models \overset{\rightsquigarrow}{\triangledown}_{ii} T^{ii}$ and T^{ii} is monotone, by Lemma 7.7

$$R_i^{AC} \text{ is monotone when } i \in A \cup C \tag{8.5}$$

It is easy to see that

$$R^{AC} \models \left(\bigwedge_{i \in A \cup C, j \in Ags} \overset{\rightsquigarrow}{\Diamond}_{ij} T^{ij} \wedge \bigwedge_{i \in (Ags \setminus (A \cup C)), j \in Ags} \overset{\rightsquigarrow}{\triangledown}_{ij} Oral(i,j) \right)$$

And (see the brief discussion on page 95)

$$R^{AC} \models \langle\langle \emptyset \rangle\rangle \Box \left(\bigwedge_{i \in A \cup C, j \in Ags} \overset{\rightsquigarrow}{\Diamond}_{ij} T^{ij} \wedge \bigwedge_{i \in (Ags \setminus (A \cup C)), j \in Ags} \overset{\rightsquigarrow}{\triangledown}_{ij} Oral(i,j) \right) \tag{8.6}$$

Similarly for R^{AB} and R^{BC}:

$$R_i^{AB} \text{ is monotone when } i \in A \cup B \tag{8.7}$$

$$R_i^{BC} \text{ is monotone when } i \in B \cup C \tag{8.8}$$

$$R^{AB}, (\vec{s}, \pi) \models \overset{\rightsquigarrow}{\Diamond}_{ij} T^{ij} \text{ } (\vec{s}, \pi, j \text{ arbitrary}, i \in A \cup B) \tag{8.9}$$

$$R^{BC}, (\vec{s}, \pi) \models \overset{\rightsquigarrow}{\Diamond}_{ij} T^{ij} \text{ } (\vec{s}, \pi, j \text{ arbitrary}, i \in B \cup C) \tag{8.10}$$

$$R^{AB} \models \langle\langle \emptyset \rangle\rangle \Box \left(\bigwedge_{i \in A \cup B, j \in Ags} \overset{\rightsquigarrow}{\Diamond}_{ij} T^{ij} \wedge \bigwedge_{i \in (Ags \setminus (A \cup B)), j \in Ags} \overset{\rightsquigarrow}{\triangledown}_{ij} Oral(i,j) \right) \tag{8.11}$$

$$R^{BC} \models \langle\langle \emptyset \rangle\rangle \Box \left(\bigwedge_{i \in B \cup C, j \in Ags} \overset{\rightsquigarrow}{\Diamond}_{ij} T^{ij} \wedge \bigwedge_{i \in (Ags \setminus (B \cup C)), j \in Ags} \overset{\rightsquigarrow}{\triangledown}_{ij} Oral(i,j) \right) \tag{8.12}$$

Note that the interpretation of the rule term $Oral(i,j)$ does not depend on the state (because nothing in the antecedent appears in the consequent of any of the rules); I henceforth write $[\![Oral(i,j)]\!]$ for $[\![Oral(i,j)]\!](s)$. To see that the fact that the T^{ij}s are oral implies that

$$[\![T^{i1}]\!](s) \times \cdots \times [\![T^{in}]\!](s) \subseteq [\![Oral(i,1)]\!] \times \cdots \times [\![Oral(i,n)]\!] \tag{8.13}$$

for any s, let R be such that $R_1(s) = [\![T^{i1}]\!](s) \times \cdots \times [\![T^{in}]\!](s)$ for all s. $R, M \models \bigwedge_{j\in[1,n]} \overset{\rightsquigarrow}{\bigtriangledown}_{ij} T^{ij}$ for all M, so $R, M \models \bigwedge_{j\in[1,n]} \overset{\rightsquigarrow}{\bigtriangledown}_{ij} Oral(i,j)$ for all M by the definition of oral rule terms. If $(s'_1, \ldots, s'_n) \in [\![T^{i1}]\!](s) \times \cdots \times [\![T^{in}]\!](s)$ for some s, then $(s'_1, \ldots, s'_n) \in R_i(s)$ and by Lemma 7.3.2 $s'_j \in [\![Oral(i,j)]\!]$ since $R, M \models \overset{\rightsquigarrow}{\bigtriangledown}_{ij} Oral(i,j)$ for some M with state s for i. Thus, $(s'_1, \ldots, s'_n) \in [\![Oral(i,1)]\!] \times \cdots \times [\![Oral(i,n)]\!]$ and (8.13) holds.

(8.13) implies that

$$R_i^{AC}(s) \subseteq [\![Oral(i,1)]\!] \times \cdots \times [\![Oral(i,n)]\!] \text{ for all } s \text{ and all } i \in A \cup C \tag{8.14}$$
$$R_i^{AB}(s) \subseteq [\![Oral(i,1)]\!] \times \cdots \times [\![Oral(i,n)]\!] \text{ for all } s \text{ and all } i \in A \cup B \tag{8.15}$$
$$R_i^{BC}(s) \subseteq [\![Oral(i,1)]\!] \times \cdots \times [\![Oral(i,n)]\!] \text{ for all } s \text{ and all } i \in B \cup C \tag{8.16}$$

I now construct vectors of epistemic states. Let $\vec{s}^{AC}, s^{AB}$ and $\vec{s}^{BC}$ be defined by the following components:

$$s_i^{AC} = \begin{cases} \{\triangle_{cg}\{attack\}, \triangle_{cg}\{\neg\neg attack\}\} & i = cg \\ \emptyset & \text{otherwise} \end{cases} \tag{8.17}$$

$$s_i^{AB} = \begin{cases} \{\triangle_{cg}\{attack\}, \triangle_{cg}\{\neg\neg attack\}\} & i = cg \\ \emptyset & \text{otherwise} \end{cases} \tag{8.18}$$

$$s_i^{BC} = \begin{cases} \{\neg \triangle_{cg} \{attack\}, \triangle_{cg}\{\neg attack\}\} & i = cg \\ \emptyset & \text{otherwise} \end{cases} \tag{8.19}$$

Clearly $R^{AC}, (\vec{s}^{AC}, \pi) \models \Gamma$, $R^{AB}, (\vec{s}^{AB}, \pi) \models \Gamma$ and $R^{BC}, (\vec{s}^{BC}, \pi) \models \Gamma$ (π arbitrary) and it follows from (8.3) and (8.6), (8.11) and (8.12) that

$$R^{AC}, (\vec{s}^{AC}, \pi) \models \langle\langle A \cup C \rangle\rangle \mathcal{F} IC(A \cup C) \tag{8.20}$$
$$R^{AB}, (\vec{s}^{AB}, \pi) \models \langle\langle A \cup B \rangle\rangle \mathcal{F} IC(A \cup B) \tag{8.21}$$
$$R^{BC}, (\vec{s}^{BC}, \pi) \models \langle\langle B \cup C \rangle\rangle \mathcal{F} IC(B \cup C) \tag{8.22}$$

Thus, there exist

$$\vec{f}^{AC} \in Str(A \cup C, R^{AC}) \tag{8.23}$$
$$\vec{f}^{AB} \in Str(A \cup B, R^{AB}) \tag{8.24}$$
$$\vec{f}^{BC} \in Str(B \cup C, R^{BC}) \tag{8.25}$$

such that, for arbitrary π and computations $\lambda_1, \lambda_2, \lambda_3$:

$$\lambda_1 \in out_{R^{AC}}(\vec{f}^{AC}, \vec{s}^{AC}) \Rightarrow \exists_{k_1} R^{AC}, (\lambda_1[k_1], \pi) \models IC(A \cup C) \tag{8.26}$$
$$\lambda_2 \in out_{R^{AB}}(\vec{f}^{AB}, \vec{s}^{AB}) \Rightarrow \exists_{k_2} R^{AB}, (\lambda_2[k_2], \pi) \models IC(A \cup B) \tag{8.27}$$
$$\lambda_3 \in out_{R^{BC}}(\vec{f}^{BC}, \vec{s}^{BC}) \Rightarrow \exists_{k_3} R^{BC}, (\lambda_3[k_3], \pi) \models IC(B \cup C) \tag{8.28}$$

Computations $\lambda_1, \lambda_2, \lambda_3$ are defined as follows. Recall that $\lambda[j]^k$ denotes the kth component of $\lambda[j]$.

- $\lambda_1[0] = \vec{s}^{AC}$ $\quad \lambda_2[0] = \vec{s}^{AB}$ $\quad \lambda_3[0] = \vec{s}^{BC}$
- $\lambda_1[j+1] = (s_1, \ldots, s_n)$ where $s_i = \cup_{k=1}^{n} s_i^k$ and
  - $k \in A \cup C$: $(s_1^k, \ldots, s_n^k) = f_k^{AC}(\lambda_1[j]^k)$
  - $k \in B$: $(s_1^k, \ldots, s_n^k) = f_k^{AB}(\lambda_2[j]^k)$
- $\lambda_2[j+1] = (s_1, \ldots, s_n)$ where $s_i = \cup_{k=1}^{n} s_i^k$ and
  - $k \in A \cup B$: $(s_1^k, \ldots, s_n^k) = f_k^{AB}(\lambda_2[j]^k)$
  - $k \in C$:
    * $a \in A$: $s_a^k = f_k^{AC}(\lambda_1[j]^k)^a$
    * $b \in B$: $s_b^k = f_k^{BC}(\lambda_3[j]^k)^b$
    * $c \in C$: $s_c^k = \emptyset$
- $\lambda_3[j+1] = (s_1, \ldots, s_n)$ where $s_i = \cup_{k=1}^{n} s_i^k$ and
  - $k \in B \cup C$: $(s_1^k, \ldots, s_n^k) = f_k^{BC}(\lambda_3[j]^k)$
  - $k \in A$: $(s_1^k, \ldots, s_n^k) = f_k^{AB}(\lambda_2[j]^k)$

I now show that

$$\lambda_1 \in out_{R^{AC}}(\vec{f}^{AC}, \vec{s}^{AC}) \qquad \lambda_2 \in out_{R^{AB}}(\vec{f}^{AB}, \vec{s}^{AB}) \qquad \lambda_3 \in out_{R^{BC}}(\vec{f}^{BC}, \vec{s}^{BC}) \tag{8.29}$$

(see the definition of *out* on p. 95).

- $\lambda_1 \in out_{R^{AC}}(\vec{f}^{AC}, \vec{s}^{AC})$:
  1. $\lambda_1[0] = \vec{s}^{AC}$.
  2. Let $j \geq 0$. $\lambda_1[j+1] = (\cup_{i=1}^{n} s_1^i, \ldots, \cup_{j=i}^{n} s_n^i)$.
     - Let $i \in [1, n]$. If $i \in A \cup C$, $(s_1^i, \ldots, s_n^i) = f_i^{AC}(\lambda_1[j]^i)$ and thus $(s_1^i, \ldots, s_n^i) \in R_i^{AC}(\lambda_1[j]^i)$ since $\vec{f}^{AC} \in Str(G, R^{AC})$. If $i \in B$, $(s_1^i, \ldots, s_n^i) = f_i^{AB}(\lambda_2[j]^i) \in R_i^{AB}(\lambda_2[j]^i) \subseteq [\![Oral(i,1)]\!] \times \cdots \times [\![Oral(i,n)]\!] = R_i^{AC}(s)$ by (8.15).
     - Let $i \in A \cup C$. $(s_1^i, \ldots, s_n^i) = f_i^{AC}(\lambda_1[j]^i)$.
- $\lambda_2 \in out_{R^{AB}}(\vec{f}^{AB}, \vec{s}^{AB})$:
  1. $\lambda_2[0] = \vec{s}^{AB}$.
  2. Let $j \geq 0$. $\lambda_2[j+1] = (\cup_{i=1}^{n} s_1^i, \ldots, \cup_{j=i}^{n} s_n^i)$.
     - Let $i \in [1, n]$. If $i \in A \cup B$, $(s_1^i, \ldots, s_n^i) = f_i^{AB}(\lambda_2[j]^i)$ and thus $(s_1^i, \ldots, s_n^i) \in R_i^{AB}(\lambda_2[j]^i)$ since $\vec{f}^{AB} \in Str(G, R^{AB})$. If $i \in C$: for $a \in A$, $s_a^i = f_i^{AC}(\lambda_1[j]^i)^a \in R_i^{AC}(\lambda_1[j]^i)^a \subseteq [\![Oral(i,a)]\!]$ by (8.14); for $b \in B$, $s_b^i = f_i^{BC}(\lambda_3[j]^i)^b \in R_i^{BC}(\lambda_3[j]^i)^b \subseteq [\![Oral(i,b)]\!]$ by (8.16); for $c \in C$, $s_c^i = \emptyset \in [\![Oral(i,c)]\!]$ by definition of $Oral(i,c)$; thus $(s_1^i, \ldots, s_n^i) \subseteq [\![Oral(i,1)]\!] \times \cdots \times [\![Oral(i,n)]\!] = R_i^{AB}(s)$ when $i \in C$.

    - Let $i \in A \cup B$. $(s_1^i, \ldots, s_n^i) = f_i^{AB}(\lambda_2[j]^i)$.

- $\lambda_3 \in out_{R^{BC}}(\vec{f}^{BC}, \vec{s}^{BC})$:
  1. $\lambda_1[0] = \vec{s}^{BC}$.
  2. Let $j \geq 0$. $\lambda_3[j+1] = (\cup_{i=1}^n s_1^i, \ldots, \cup_{j=i}^n s_n^i)$.
     - Let $i \in [1, n]$. If $i \in B \cup C$, $(s_1^i, \ldots, s_n^i) = f_i^{BC}(\lambda_3[j]^i)$ and thus $(s_1^i, \ldots, s_n^i) \in R_i^{BC}(\lambda_3[j]^i)$ since $\vec{f}^{BC} \in Str(G, R^{BC})$. If $i \in A$, $(s_1^i, \ldots, s_n^i) = f_i^{AB}(\lambda_2[j]^i) \in R_i^{AB}(\lambda_2[j]^i) \subseteq [\![Oral(i,1)]\!] \times \cdots \times [\![Oral(i,n)]\!] = R_i^{BC}(s)$ by (8.15).
     - Let $i \in B \cup C$. $(s_1^i, \ldots, s_n^i) = f_i^{BC}(\lambda_3[j]^i)$.

The following states that the computations λ_1 and λ_2 are equal for all the agents in A:

$$\text{For all } k \text{ and all } a \in A\text{: } \lambda_1[k]^a = \lambda_2[k]^a \tag{8.30}$$

I show (8.30) by induction over k.

- $k = 0$: Let $a \in A$. $\lambda_1[0]^a = s_a^{AC} = \emptyset = s_a^{AB} = \lambda_2[0]^a$ (since $cg \notin A$).
- $k = k' + 1$: Let $a \in A$. $\lambda_1[k'+1]^a = \cup_{j=1}^n s_a^j$ and $\lambda_2[k'+1]^a = \cup_{j=1}^a \tilde{s}_a^j$. I show that $s_a^j = \tilde{s}_a^j$ for every $j \in [1, n]$:
  - $j \in B$: $s_a^j = f_j^{AB}(\lambda_2[k']^j)^a = \tilde{s}_a^j$.
  - $j \in C$: $s_a^j = f_j^{AC}(\lambda_1[k']^j)^a = \tilde{s}_a^j$.
  - $j \in A$: $(s_1^j, \ldots, s_n^j) = f_j^{AC}(\lambda_1[k']^j) \in R_a^{AC}(\lambda_1[k']^a)$. $(\tilde{s}_1^j, \ldots, \tilde{s}_n^j) = f_j^{AB}(\lambda_2[k']^j) \in R_a^{AB}(\lambda_2[k']^a)$. By the induction hypothesis $\lambda_1[k']^a = \lambda_2[k']^a = s$. Since $a \in A$, $R_a^{AC}(s) = [\![T^{a1}]\!](s) \times \cdots [\![T^{an}]\!](s) = R_a^{AB}(s)$. Because all T^{aj} are deterministic, each $[\![T^{aj}]\!]$ is singular, and thus $f_j^{AC}(\lambda_1[k']^j) = f_j^{AB}(\lambda_2[k']^j)$. $s_a^j = f_j^{AC}(\lambda_1[k']^j)^a = f_j^{AB}(\lambda_2[k']^j)^a = \tilde{s}_a^j$.'

Similarly, the computations λ_2 and λ_3 are equal for all the agents in B:

$$\text{For all } k \text{ and all } b \in B\text{: } \lambda_2[k]^b = \lambda_3[k]^b \tag{8.31}$$

The proof is completely symmetric to the proof of (8.30):

- $k = 0$: Let $b \in B$. $\lambda_2[0]^b = s_b^{AB} = \emptyset = s_b^{BC} = \lambda_3[0]^b$ (since $cg \notin B$).
- $k = k' + 1$: Let $b \in B$. $\lambda_2[k'+1]^b = \cup_{j=1}^n s_b^j$ and $\lambda_3[k'+1]^b = \cup_{j=1}^b \tilde{s}_b^j$. I show that $s_b^j = \tilde{s}_b^j$ for every $j \in [1, n]$:
  - $j \in A$: $s_b^j = f_j^{AB}(\lambda_2[k']^j)^b = \tilde{s}_b^j$.
  - $j \in C$: $s_b^j = f_j^{BC}(\lambda_3[k']^j)^b = \tilde{s}_a^j$.

- $j \in B$: $(s_1^j, \ldots, s_n^j) = f_j^{AB}(\lambda_2[k']^j) \in R_b^{AB}(\lambda_2[k']^b)$. $(s_1^{\tilde{j}}, \ldots, s_n^{\tilde{j}}) = f_j^{BC}(\lambda_3[k']^j) \in R_b^{BC}(\lambda_3[k']^b)$. By the induction hypothesis $\lambda_2[k']^b = \lambda_3[k']^b = s$. Since $b \in B$, $R_b^{AB}(s) = [\![T^{b1}]\!](s) \times \cdots [\![T^{bn}]\!](s) = R_a^{BC}(s)$. Because all T^{bj} are deterministic, each $[\![T^{bj}]\!]$ is singular, and thus $f_j^{AB}(\lambda_2[k']^j) = f_j^{BC}(\lambda_3[k']^j)$. $s_b^j = f_j^{AB}(\lambda_2[k']^j)^b = f_j^{BC}(\lambda_3[k']^j)^b = s_b^{\tilde{j}}$.

Since λ_1 is a R^{AC}-computation and R_i^{AC} is monotone when $i \in A \cup C$ (8.5),

$$\lambda_1[k]^i \subseteq \lambda_1[k+1]^i \text{ when } i \in A \cup C \tag{8.32}$$

for all $k \geq 0$ by Lemma 7.6. Similarly,

$$\lambda_2[k]^i \subseteq \lambda_2[k+1]^i \text{ when } i \in A \cup B \tag{8.33}$$

$$\lambda_3[k]^i \subseteq \lambda_3[k+1]^i \text{ when } i \in B \cup C \tag{8.34}$$

Finally, the computations λ_1, λ_2 and λ_3 are used to show a contradiction. Let $a \in A$ and $b \in B$, and let π be arbitrary. (Note that some of the following facts are named by equation numbers in single parentheses in the rightmost column, and the middle column is used for justification).

$$R^{AC},(\lambda_1[0],\pi) \models \triangle_{cg}\{\triangle_{cg}\{attack\}\} \qquad (\lambda_1[0]=\vec{s}^{AC})$$
$$R^{AC},(\lambda_1[k_1],\pi) \models \triangle_{cg}\{\triangle_{cg}\{attack\}\} \qquad ((8.32), cg \in A\cup C)$$
$$R^{AC},(\lambda_1[k_1],\pi) \models IC(A\cup C) \qquad ((8.26))$$
$$R^{AC},(\lambda_1[k_1],\pi) \models \triangle_{cg}\{\triangle_{cg}\{attack\}\} \leftrightarrow \triangle_a\{\triangle_{cg}\{attack\}\} \qquad (a, cg \in A\cup C)$$
$$R^{AC},(\lambda_1[k_1],\pi) \models \triangle_a\{\triangle_{cg}\{attack\}\} \leftrightarrow \neg\triangle_a\{\neg\triangle_{cg}\{attack\}\} \qquad (a \in A\cup C)$$
$$R^{AC},(\lambda_1[k_1],\pi) \models \triangle_a\{\triangle_{cg}\{attack\}\}$$
$$R^{AC},(\lambda_1[k_1],\pi) \models \neg\triangle_a\{\neg\triangle_{cg}\{attack\}\}$$
$$\triangle_{cg}\{attack\} \in \lambda_1[k_1]^a \qquad (8.35)$$
$$\neg\triangle_{cg}\{attack\} \notin \lambda_1[k_1]^a \qquad (8.36)$$
$$R^{BC},(\lambda_3[0],\pi) \models \triangle_{cg}\{\neg\triangle_{cg}\{attack\}\} \qquad (\lambda_3[0]=\vec{s}^{BC})$$
$$R^{BC},(\lambda_3[k_3],\pi) \models \triangle_{cg}\{\neg\triangle_{cg}\{attack\}\} \qquad ((8.33), cg \in B\cup C)$$
$$R^{BC},(\lambda_3[k_3],\pi) \models IC(B\cup C) \qquad ((8.28))$$
$$R^{BC},(\lambda_3[k_3],\pi) \models \triangle_{cg}\{\neg\triangle_{cg}\{attack\}\} \leftrightarrow \triangle_b\{\neg\triangle_{cg}\{attack\}\} \qquad (b, cg \in B\cup C)$$
$$R^{BC},(\lambda_3[k_3],\pi) \models \triangle_b\{\triangle_{cg}\{attack\}\} \leftrightarrow \neg\triangle_b\{\neg\triangle_{cg}\{attack\}\} \qquad (b \in B\cup C)$$
$$R^{BC},(\lambda_3[k_3],\pi) \models \triangle_b\{\neg\triangle_{cg}\{attack\}\}$$
$$R^{BC},(\lambda_3[k_3],\pi) \models \neg\triangle_b\{\triangle_{cg}\{attack\}\}$$
$$\neg\triangle_{cg}\{attack\} \in \lambda_3[k_3]^b \qquad (8.37)$$
$$\triangle_{cg}\{attack\} \notin \lambda_3[k_3]^b \qquad (8.38)$$
$$R^{AB},(\lambda_2[k_2],\pi) \models IC(A\cup B) \qquad ((8.27))$$
$$R^{AB},(\lambda_2[k_2],\pi) \models \triangle_a\{\triangle_{cg}\{attack\}\} \leftrightarrow \triangle_b\{\triangle_{cg}\{attack\}\} \qquad (a, b \in A\cup B)$$
$$R^{AB},(\lambda_2[k_2],\pi) \models \triangle_a\{\triangle_{cg}\{attack\}\} \vee \triangle_a\{\neg\triangle_{cg}\{attack\}\} \qquad (a \in A\cup B)$$
$$R^{AB},(\lambda_2[k_2],\pi) \models \triangle_b\{\triangle_{cg}\{attack\}\} \vee \triangle_b\{\neg\triangle_{cg}\{attack\}\} \qquad (b \in A\cup B)$$
$$R^{AB},(\lambda_2[k_2],\pi) \models \triangle_b\{\neg\triangle_{cg}\{attack\}\} \leftrightarrow \neg\triangle_b\{\triangle_{cg}\{attack\}\} \qquad (b \in A\cup B)$$
$$\triangle_{cg}\{attack\} \in \lambda_2[k_2]^a \Leftrightarrow \triangle_{cg}\{attack\} \in \lambda_2[k_2]^b \qquad (8.39)$$
$$\triangle_{cg}\{attack\} \in \lambda_2[k_2]^a \text{ or } \neg\triangle_{cg}\{attack\} \in \lambda_2[k_2]^a \qquad (8.40)$$
$$\triangle_{cg}\{attack\} \in \lambda_2[k_2]^b \text{ or } \neg\triangle_{cg}\{attack\} \in \lambda_2[k_2]^b \qquad (8.41)$$
$$\neg\triangle_{cg}\{attack\} \in \lambda_2[k_2]^b \Rightarrow \triangle_{cg}\{attack\} \notin \lambda_2[k_2]^b \qquad (8.42)$$
$$\triangle_{cg}\{attack\} \in \lambda_2[k_1]^a \qquad ((8.30),(8.35)) \quad (8.43)$$

If $k_2 \geq k_1$ then $\lambda_2[k_1]^a \subseteq \lambda_2[k_2]^a$ by monotonicity of R_a^{AB}. If $k_2 < k_1$ then $\neg\triangle_{cg}\{attack\} \notin \lambda_2[k_2]^a$ because if $\neg\triangle_{cg}\{attack\} \in \lambda_2[k_2]^a$ then $\neg\triangle_{cg}\{attack\} \in \lambda_1[k_2]^a$ by eq. (8.30) and $\neg\triangle_{cg}\{attack\} \in \lambda_1[k_1]^a$ by monotonicity of R_a^{AC} which is a contradiction by eq. 8.36. By 8.40, $\triangle_{cg}\{attack\} \in \lambda_2[k_2]^a$. In either case:

$$\triangle_{cg}\{attack\} \in \lambda_2[k_2]^a \tag{8.44}$$

$$\triangle_{cg}\{attack\} \in \lambda_2[k_2]^b((8.39)) \tag{8.45}$$

$$\neg \triangle_{cg} \{attack\} \in \lambda_2[k_3]^b((8.37),(8.31)) \tag{8.46}$$

By the same reasoning as above: If $k_2 \geq k_3$ then $\lambda_2[k_3]^b \subseteq \lambda_2[k_2]^b$ by monotonicity of R_b^{AB}. If $k_2 < k_3$ then $\triangle_{cg}\{attack\} \notin \lambda_2[k_2]^b$ because if $\triangle_{cg}\{attack\} \in \lambda_2[k_2]^b$ then $\triangle_{cg}\{attack\} \in \lambda_3[k_2]^b$ by eq. (8.31) and $\triangle_{cg}\{attack\} \in \lambda_3[k_3]^b$ by monotonicity of R_b^{BC} which is a contradiction by eq. 8.38. By (8.41), $\neg \triangle_{cg}$ $\{attack\} \in \lambda_2[k_2]^b$. In either case:

$$\neg \triangle_{cg} \{attack\} \in \lambda_2[k_2]^b \tag{8.47}$$

$$\triangle_{cg}\{attack\} \notin \lambda_2[k_2]^b(eq.(8.42)) \tag{8.48}$$

which is a contradiction by eq. (8.45).

Thus, the theorem must be true. ■

8.4 A Possibility Result

If the traitors are restricted to oral messages, there exists a protocol the agents can use successfully if *more* than two thirds of the generals are loyal.

The result in this section is a formulation of a result from Lamport, Shostak, & Pease (1982). The presentation differs from the one of the negative result in the previous section in two important ways. First, the formulation of the main result, Theorem 8.2, is not on the form of Theorem 8.1 but has a more meta logical flavour. Second, the proof of the result, which again is an adaption of a known proof, and is found in an appendix, is not presented as rigorously as the proof of the impossibility result in the previous section. The reason for these differences is that proving validity in the presented semantics for DSEL can be very extensive and technical (see e.g. the proof of the small modus ponens result in Example 7.2 on page 99), and the model of the problem is quite complex.

In the following theorem, R is a general mechanism which allows agents to do any reasoning actions, modeled by $\overset{\leadsto}{\Diamond}_{ii}\{\frac{t}{u}\}$ (recall that t, u are term variable), and any communication actions restricted to oral messages. A *monotone strategy* is a strategy $\vec{f}$ where $s \subseteq f_i(s)^i$ for each i and each s. Γ was defined in Def. 8.2 on p. 110.

Theorem 8.2 Let

$$R \models \bigwedge_{i \in Ags} \overset{\leadsto}{\Diamond}_{ii}\{\frac{t}{u}\} \wedge \bigwedge_{i \neq j \in Ags} \overset{\leadsto}{\Diamond}_{ij} Oral(i,j)$$

(note that R is unique by Lemma 7.3.4). Let

$$R, (\vec{s}, \pi) \models \Gamma$$

There exists monotone $\vec{f}_{Ags} \in Str(Ags, R)$ such that for all $G \subseteq Ags$ with $|G| > \frac{2n}{3}$, for all $\lambda \in out_R(\vec{f}_G, \vec{s})$ there exists a k such that

$$R, (\lambda[k], \pi) \models IC(G) \qquad \square$$

PROOF Appendix D. ■

$\vec{f}_G \subseteq \vec{f}_{Ags}$ are the strategies in $\vec{f}_{Ags}$ only for $G \subseteq Ags$.

As mentioned, this result is more meta logical than the result in the previous section; it uses the semantical concept of strategies instead of the syntactical concept of rules. Theorem 8.2 essentially says that the loyal agents, if many enough, have a *strategy* to achieve interactive consistency. If that strategy can be described by a set of (deterministic) rules, the reverse of the impossibility result holds, as shown in the following subsection.

8.4.1 Rules

Conjecture 8.1 Let $\vec{f}_{Ags}$ be as defined in Th. 8.2. There exist T^{ij}, $i, j \in [1, n]$, such that for all s:

$$[\![T^{ij}]\!](s) = \{f_i(s)^j\} \qquad \square$$

Corollary 8.1 (Based on Conjecture 8.1) There exist a combination of complete and deterministic rules T^{ij} for every agent i and j such that T^{ii} is monotone for each i, T^{ij} is oral for each $i \neq j$ and for every $G \subseteq Ags$ such that $|G| > \frac{2n}{3}$

$$\models \langle\langle\emptyset\rangle\rangle\Box\left(\bigwedge_{i\in G, j\in Ags} \overset{\leadsto}{\Diamond}_{ij} T^{ij} \wedge \bigwedge_{i\in(Ags\setminus G), i\in Ags} \overset{\leadsto}{\nabla}_{ij} Oral(i,j)\right) \rightarrow (\Gamma \rightarrow \langle\langle G\rangle\rangle \mathcal{F} IC(G)) \tag{8.49}$$

$\square$

PROOF Let T^{ij} be as in Conjecture 8.1; each T^{ij} is clearly both complete and deterministic, T^{ii} is monotone since f_i is monotone and T^{ij}, $i \neq j$, is oral since f_i is restricted to oral messages. Let $G \subseteq Ags$ such that $|G| > \frac{2n}{3}$, and let

$$R', (\vec{s}', \pi') \models \langle\langle\emptyset\rangle\rangle\Box\left(\bigwedge_{i\in G, j\in Ags} \overset{\leadsto}{\Diamond}_{ij} T^{ij} \wedge \bigwedge_{i\in(Ags\setminus G), i\in Ags} \overset{\leadsto}{\nabla}_{ij} Oral(i,j)\right) \wedge \Gamma$$

Let $R, \vec{s}$ and $\vec{f}_{Ags}$ be as defined in Th 8.2. By construction of Γ, $\vec{s}' = \vec{s}$. Let $\vec{f}'_G$ be defined as follows: if $s = \lambda[p]^i$ for any R'-computation λ with $\lambda[0] = \vec{s}$ and any p then $f'_i(s) = f_i(s)$, otherwise $f'_i(s)$ is arbitrary such that $f'_i(s) \in R'_i(s)$ ($R'_i(s) \neq \emptyset$). $\vec{f}_G \in Str(G, R)$. To show that $\vec{f}'_G \in Str(G, R')$, we must show that $\vec{f}'_i(s) \in R'_i(s)$ for all s and $i \in G$. Let $i \in G$. If $s = \lambda[p]^i$ for a R'-computation λ with $\lambda[0] = \vec{s}$ and some p (otherwise $\vec{f}'_i(s) \in R'_i(s)$ trivially), then, since $R', (\lambda[p], \pi') \models \wedge_{j\in Ags} \overset{\leadsto}{\Diamond}_{ij} T^{ij}$, $R'_i(s) = [\![T^{i1}]\!](s) \times \cdots \times [\![T^{in}]\!](s)$ by Lemma 7.3.4. By assumption of T^{ij}, $R'_i(s) = \{(f_i(s)^1, \ldots, f_i(s)^n)\}$, so $f'_i(s) = f_i(s) \in R'_i(s)$. Thus, $\vec{f}'_G \in Str(G, R')$.

Let $\lambda \in out_{R'}(\vec{f}'_G, \vec{s})$. Since $\vec{f}'_G$ and $\vec{f}_G$ agrees for all R'-computations starting in $\vec{s}$ and $R'_i(\lambda[p]^i) \subseteq R_i(\lambda[p]^i)$ (apply Lemma 7.3, points 2 and 4) for all $i \in Ags \setminus G$ and all p, it is easy to see that $\lambda \in out_R(\vec{f}_G, \vec{s})$. Since $R, (\lambda[k], \pi) \models IC(G)$, $R', (\lambda[k], \pi') \models IC(G)$ ($IC(G)$ does not depend on R or π), and thus

$$R', (\vec{s}', \pi') \models \langle\langle G \rangle\rangle \mathcal{F} IC(G)$$ ■

Part IV

Discussion and Conclusions

Chapter 9

Comparison to Earlier Work

9.1 Introduction

The work presented in Parts II and III of this thesis can respectively be compared to different kinds of earlier work.

The idea of representing knowledge syntactically is of course not new, and in Section 9.2 the theory from Part II is compared to earlier work. The concept of "only knowing" was a motivation for the $\bigtriangledown_i$ operator, and is briefly reviewed from the literature in Section 9.3.

The rest of the sections in this chapter compare models of knowledge evolving over time as a result of reasoning and communication with DSEL from Part III. The literature here is quite extensive, so only a few selected approaches are discussed. One approach close to DSEL is Alternating-time Temporal Epistemic Logic (ATEL), which is another integration of ATL with epistemic logic — albeit with implicit knowledge. ATEL is discussed in Section 9.4. In Section 9.5 DSEL is compared with the extensive framework for modeling interpreted systems presented in the book "Reasoning about Knowledge" (Fagin *et al.*, 1995). Instead of using a temporal logic to model reasoning, dynamic logic can be used. One such approach is discussed in Section 9.6. The last section briefly discusses the extensive first order framework called Active Logics (formerly "Step Logics").

Among the approaches not discussed further are the following. In Konolige's *deduction model* (Konolige, 1984, 1985, 1986a) agents are modeled by a belief set and a set of deduction rules where the former is closed under the latter rather than under logical consequence and full logical omniscience is avoided by using incomplete deduction rules[1]. Active Logics provide a similar model which is more fine grained. Moses (1988), Halpern, Moses, & Vardi (1994) and Duc (1997a) present different approaches to reasoning bounded by the algorithms available to the agent and/or the complexity of reasoning.

[1]Wooldridge (1995a) generalizes this model by replacing deduction rules with a binary relation.

9.2 Syntactic Representations

Syntactic approaches are suggested by Eberle (1974), and Moore & Hendrix (1979). Halpern & Moses (1990) present such an approach within the possible worlds framework as a general model for knowledge in distributed systems, suggesting that a general model is needed for, for example, cases where knowledge is dependent on the processors' computational powers. The following presentation is taken from Fagin *et al.* (1995). The idea is, for each state, to assign a truth value to *any* formula independently. A *syntactic structure* M is a pair (S, σ), where S is a set of states and σ is a *standard syntactic assignment*. A standard syntactic assignment σ gives a mapping $\sigma(s)$ for each state $s \in S$ from the set of well-formed formulae to the set $\{\mathbf{true}, \mathbf{false}\}$. Constraints are imposed on the standard syntactic assignment to preserve the standard propositional semantics. For all standard syntactic assignments σ and well-formed formulae ϕ and ψ:

$$\sigma(s)(\phi) = \mathbf{true} \Leftrightarrow \sigma(s)(\neg\phi) = \mathbf{false} \tag{9.1}$$

$$\begin{aligned}\sigma(s)(\phi \wedge \psi) = \mathbf{true} \Leftrightarrow \sigma(s)(\phi) &= \mathbf{true} \text{ and} \\ \sigma(s)(\psi) &= \mathbf{true}\end{aligned} \tag{9.2}$$

A well-formed formula ϕ is true in a state s in the syntactic structure M, written $(M, s) \models \phi$ if and only if $\sigma(s)(\phi) = \mathbf{true}$. Syntactic structures are generalizations of Kripke structures[2].

9.2.1 Comparison

Clearly, syntactic structures are closely related to the framework for syntactic knowledge presented in Part II. Although agents described by syntactic structures are not partially omniscient, they can have infinite knowledge.

It is easy to see that GKSSs (KSSs possibly with infinite knowledge) are at least[3] equivalent to syntactic structures or, alternatively, that KSSs are equivalent to syntactic structures restricted to assigning finite knowledge to each agent, in the following way. If we define satisfiability of $\mathcal{L}_n(\Phi)$ (see p. 9) in a (G)KSS by interpreting the K_i operator as the $\triangle_i$ operator, then for every syntactic structure $M = (S, \sigma)$ and state $s \in S$ there is a (G)KSS $M' = (s_1, \ldots, s_n, \pi)$ such that[4] for any $\phi \in \mathcal{L}_n(\Phi)$

$$M, s \models \phi \Leftrightarrow M' \models \phi$$

[2]As pointed out by Fagin *et al.* (1995), syntactic structures are also generalizations of *Montague-Scott (MS) structures* (Montague, 1968, 1970; Scott, 1970). An MS structure is a tuple $M = (S, \pi, \mathcal{C}_1, \ldots, \mathcal{C}_n)$, where S is a set of states, $\pi(s)$ is a truth assignment to the primitive propositions in each state s, and $\mathcal{C}_i(s)$ $(1 \leq i \leq n)$ is a set of subsets of S representing the *intentions* of the propositions agent i knows in state s. $(M, s) \models K_i\phi$ if and only if $\{s' : (M, s') \models \phi\} \in \mathcal{C}_i(s)$. Unlike in syntactic structures, knowledge is closed under logical equivalence.

Vardi (1986) finds MS structures unsatisfying because it is not clear what the possible worlds are, and proposes a constructive definition of *belief worlds* (called *knowledge structures* by Fagin, Halpern, & Vardi (1991)) instead.

[3]GKSSs are strictly more general, because of the $*$ element.

[4]Note that the definition of satisfiability with respect to two components, a "structure" and a "state", for syntactic structures is superficial: like for (G)KSSs satisfiability depends only on one state; the definition is not recursive.

by taking

$$\pi(p) = \sigma(s)(p)$$

$$s_i = \{\phi : \sigma(s)(K_i\phi) = \mathbf{true}\}$$

Since (G)KSSs are almost identical to syntactic structures, the possible novelty of the logic SSEL must be discussed. The main difference between the language *EL* and $\mathcal{L}_n(\Phi)$ is the $\bigtriangledown_i$ operator. It allows expressing in a single formula certain facts which would have needed *infinitely* many $\mathcal{L}_n(\Phi)$ formulae. For example, the fact that agent i does not know α for any $\alpha \neq p$ is expressed in *EL* as

$$\bigtriangledown_i\{p\}$$

but must be expressed in $\mathcal{L}_n(\Phi)$ as the infinite set of formulae

$$\{\neg K_i\alpha : \alpha \neq p\} \tag{9.3}$$

Another example of the expressiveness introduced by the $\bigtriangledown_i$ operator is the following

$$\triangle_i\{\bigtriangledown_j\{p\}\}$$

which expresses the fact that agent i knows that agent j does not know anything else than p. The intent of this formula is not expressible in $\mathcal{L}_n(\Phi)$. The reason for this is that such an expression would be on the form $K_i\phi$, where ϕ is a conjunction of the set in eq. (9.3) — but this set is infinite.

Other new aspects in SSEL as a logic of syntactic knowledge is the ability to express sets as terms in the language, a sound and complete calculus for terms, a semantics for possibly infinite agents with a sound and complete axiomatization, a finite semantics (which corresponds to finitely restricted syntactic structures), and a characterization of the finitary theories for which the axiomatization is complete w.r.t. the finite semantics which among other things can help extend the axiomatization with additional restrictions on the syntactic structures while retaining completeness.

9.2.2 The Logic of General Awareness

A well-known approach which combines a syntactical and a semantical approach should also be mentioned: *the logic of general awareness* (Fagin & Halpern, 1988). The authors argue that one source of logical non-omniscience is the lack of *awareness*, i.e. that in order to say that an agent knows a fact ϕ, 1) ϕ must follow from the agent's information and 2) it must be aware of ϕ. They use three different modalities in the language to account for these notions. The knowledge[5] operators K_i and X_i are meant to capture express implicit and explicit knowledge, respectively. The awareness operator A_i is used to express the awareness of agent i of some formula ϕ; $A_i\phi$. Fagin & Halpern (1988) do not attach any fixed meaning to this notion of awareness, but leave it open to interpretation. Their approach extends the possible worlds framework with a syntactic awareness function. Formally, a *Kripke structure for general awareness* is a tuple $M = (S, \pi, \mathcal{K}_1, \dots, \mathcal{K}_n, \mathcal{A}_1, \dots, \mathcal{A}_n)$ where $(S, \pi, \mathcal{K}_1, \dots, \mathcal{K}_n)$ is a

[5]This approach is described in terms of belief rather than knowledge in (Fagin & Halpern, 1988). The distinction is not important here.

Kripke structure[6] and A_i a function from S to the set of all sets of well-formed formulae. The set $A_i(s)$ denotes the formulae agent i is aware of. The satisfiability relation is defined as usual for atomic propositions, the propositional connectives and the (implicit) knowledge operators K_i (eqs. 2.1 – 2.4) and as follows for the new connectives:

$$(M, s) \models A_i\phi \Leftrightarrow \phi \in \mathcal{A}_i(s) \tag{9.4}$$

$$(M, s) \models X_i\phi \Leftrightarrow (M, s) \models A_i\phi \text{ and } (M, s) \models K_i\phi \tag{9.5}$$

Thus, an agent explicitly knows a fact if he implicitly knows it (it is true in all the worlds he considers possible) and he is aware of it. Note that if an agent is aware of all his implicit knowledge, explicit and implicit knowledge coincide. A sound and complete logical system is given by adding the axiom

$$X_i\phi \leftrightarrow (A_i\phi \wedge K_i\phi) \tag{9.6}$$

to a sound and complete system with respect to Kripke semantics, e.g. $S5_n$ (or $KD45_n$ for belief).

As Konolige (1986b) points out, the logic of general awareness characterizes the agents as perfect reasoners restricted to considering only a subset of all possible sentences[7]. It is like syntactic structures restricted to assign truth to formulae which actually follows. It is therefore more interesting to compare SSEL to syntactic structures than to the logic of general awareness.

9.3 Only Knowing

Several authors have analyzed the knowledge state of an agent who knows a (set of) formula(e) (Konolige, 1982; Moore, 1983; Halpern & Moses, 1985; Halpern, 1997). Levesque (1990) introduced a logic in which *only knowing* can be expressed in the logical language. Briefly speaking, Levesque's language is of first order[8] and has two unary epistemic connectives **B** and **O**.[9] Semantically, a *world* is a truth assignment to the primitive sentences, and satisfaction of a formula is defined relative to a pair W, w where W is the set of worlds the agent considers possible and w is the "real" world[10] (the world corresponding to the correct state of affairs). A sentence $\mathbf{B}\alpha$ is true in W, w iff α is true in W, w' for every $w' \in W$; **B** is the traditional belief/knowledge operator in modal epistemic logic. A sentence $\mathbf{O}\alpha$ is true in W, w iff $\mathbf{B}\alpha$ is true in W, w and $w' \in W$ for every w' such that α is true in W, w'. $\mathbf{O}\alpha$ expresses that the agent only knows α; the set of possible worlds is as large as possible consistent with believing α. It is shown that the **O** operator can be modeled by a "natural dual" to the **B** operator — an operator **N**. The intended meaning of $\mathbf{N}\alpha$ is that α at most is believed to be false, and $\mathbf{N}\alpha$ is true in W, w iff $w' \in W$ for every w' such

[6]In the paper, the relations in the structure are required to be equivalence relations (serial, transitive and Euclidean for belief). There is no reason not to consider the general case without requiring certain properties of knowledge.

[7]See also (Hadley, 1986, 1988) for a critique of the notion of "awareness" and the axiom (9.6).

[8]The logic was only shown to be complete for the unquantified version of the language, the full version was later shown to be incomplete (Halpern & Lakemeyer, 1995).

[9]Levesque only considers a single agent, but his approach has later been extended to the multi-agent case (Halpern & Lakemeyer, 1996)

[10]Note that this corresponds to the semantical assumptions of the modal logic $S5$ for one agent.

that α is false in W, w'. Then, $\mathbf{O}\alpha$ is true iff $\mathbf{B}\alpha$ and $\mathbf{N}\neg\alpha$ is true; **B** specifies a lower bound and **N** specifies an upper bound on what is believed.

Levesque's approach suffers from the logical omniscience problem. The **O** operator tells which equivalence class of formulae the agent knows; knowledge is closed under logical equivalence — a clearly unrealistic assumption. Lakemeyer (1996) combines only knowing with Levesque's concept of explicit knowledge (Levesque, 1984a) as described in Section 2.2.2. Since the latter approach is not a logic about syntactic explicit knowledge, but about a weaker closure of knowledge than logical consequence, the concept of explicitly only knowing in this context is quite different from the concept of explicitly only knowing a set of formulae ($\bigtriangledown_i X$) introduced in this thesis.

9.4 Alternating-time Temporal Epistemic Logic

In Section 2.3, ATL was presented as a generalization of CTL for multi-agent systems. van der Hoek & Wooldridge (2002) have extended ATL with epistemic modalities, into *Alternating-time Temporal Epistemic Logic (ATEL)*. The extension is twofold: concurrent game structures are extended to *alternating epistemic transition systems (AETS)*, and the language of ATEL is the language of ATL extended with epistemic operators.

An AETS is a tuple

$$(\Sigma, Q, \Pi, \pi, \delta, \sim_1, \ldots, \sim_n)$$

where

- $\Sigma = \{a_1, \ldots, a_n\}$ is a set of agents
- Q is a finite, non-empty set of states
- Π is a finite, non-empty set of propositions
- $\pi : Q \to 2^{\Pi}$
- $\delta : Q \times \Sigma \to 2^{2^Q}$; the transition function. It is required that, for every $q \in Q$, $a \in \Sigma$ and $Q_a \in \delta(q, a)$: $|\bigcap_{a \in \Sigma} Q_a| = 1$.
- For each $a \in \Sigma$, $\sim_a \subseteq Q \times Q$; the epistemic accessibility relation. $\sim_a$ is required to be an equivalence relation.

The definition of AETSs is based on a previous version (Alur, Henzinger, & Kupferman, 1999) of concurrent game structures, called *alternating transition systems (ATS)*, which is slightly different from the one presented in Section 2.3 (Alur, Henzinger, & Kupferman, 2002). The difference is that there is no d_a in ATSs, and $\delta(q, a)$ is a subset of $\wp(Q)$. The transition function in ATSs gives a set of *choices* for each agent in each state, where a choice is a set of states. The system is completely controlled by the agents, and the result of each agent making a choice is the single state in the intersection of the choices. The notions of a computation and a strategy is accordingly different: a computation is a sequence of states $q_0 q_1 \cdots$ such that for each k, for each $a \in \Sigma$ there is a $Q_a \in \delta(q_k, a)$ such that $q_{k+1} \in Q_a$, and a strategy is a function $f_a : Q^+ \to 2^Q$ such that

$f_a(\lambda q) \in \delta(q, a)$. The function *out* still gives the set of possible computations for a given set of strategies and a start state.

The extension consists of a reflexive, symmetric, and transitive accessibility relation for each agent.

The language of ATEL is the language of ATL with the following extensions:

- If ϕ is a formula and $a \in \Sigma$, then $K_a\phi$ is a formula
- If ϕ is a formula and $A \subseteq \Sigma$, then $C_A\phi$ and $E_A\phi$ are formulae.

The meaning of the three operators is (implicit) knowledge, common knowledge and "everyone knows". The definition of satisfiability of an ATL formula in a state q in a concurrent game structure is extended to satisfiability of an ATEL formula in a AETS $S = (\Sigma, Q, \Pi, \pi, \delta, \sim_1, \dots, \sim_n)$ as follows:

- $S, q \models K_a\phi$ iff for all q' such that $q \sim_a q'$: $S, q' \models \phi$
- $S, q \models E_A\phi$ iff for all q' such that $q \sim_A^E q'$: $S, q' \models \phi$
- $S, q \models C_A\phi$ iff for all q' such that $q \sim_A^C q'$: $S, q' \models \phi$

where $\sim_A^E = \bigcup_{a \in A} \sim_a$ and $\sim_A^C$ is the transitive closure of $\sim_A^E$.

Note that knowledge in ATEL has the S5 properties since the accessibility relations are equivalence relations.

9.4.1 A Problem with Incomplete Information

As pointed out by Jamroga (2003), ATEL does not seem to integrate the semantics of knowledge with the ATL semantics properly: an agent can have a strategy which gives *different* choices for *indiscernible* states. In other words, if $q, q' \in Q$, $q \neq q'$ and $q \sim_i q'$, agent i can have a strategy f_i with $f_i(\lambda q) \neq f_i(\lambda q')$ for some λ. The fact that agents can base their choices on the state of the whole system, seems to contradict the premise of epistemic logic: that agents have incomplete information.

Jamroga (2003) argues that a natural solution of the problem would be to require that a strategy should specify the same action in indiscernible states, but that this constraint is difficult to express due to the way choices are defined in AETS. This, again, is due to the subtle difference between the two definitions of concurrent game structures discussed above. Jamroga therefore proposes an improved version of an AETS which is based on the latest version of concurrent game structures (with action names instead of numbers):

$$(\Sigma, Q, \Pi, \pi, \sim_1, \dots, \sim_n, ACT, d, \delta)$$

where ACT, d and δ is as defined in Definition 7.8 (p. 89), and to restrict d such that the same actions are available in indiscernible states:

$$q \sim_i q' \Rightarrow d_i(q) = d_i(q')$$

Now, strategies can be restricted such that they specify the same action for indiscernible (histories of) states:

$$\forall_{j \in [0,k]}(q_j \sim_i q'_j) \Rightarrow f_i(q_0 \cdots q_k) = f_i(q'_0 \cdots q'_k)$$

Jamroga notes that these proposals does not solve all problems with the integration of knowledge and action in ATEL.

9.4.2 Syntactic Characterization of Incomplete Information

van der Hoek & Wooldridge (2003, p. 144) very briefly comment on the problem identified by Jamroga (2003) in a discussion on how to apply ATEL to games involving knowledge, and propose to solve it in a similar way in that specific context. They claim that the semantic requirement

$$q \sim_a q' \Rightarrow \delta(q,a) = \delta(q',a) \tag{9.7}$$

is captured by the syntactic property

$$\langle\langle\{a\}\rangle\rangle T\phi \leftrightarrow K_a\langle\langle\{a\}\rangle\rangle T\phi \tag{9.8}$$

where $T \in \{\bigcirc, \Box, \mathcal{F}\}$.

However, this claim is false. (9.8) is neither a necessary nor a sufficient condition for (9.7). The following is a counter-example for the latter. Let $\phi = p$. I show that there is an AETS S where (9.7) holds for all states q, q' and agents a, but where there is a state q_1 such that

$$S, q_1 \not\models \langle\langle\{a\}\rangle\rangle \bigcirc p \rightarrow K_a\langle\langle\{a\}\rangle\rangle \bigcirc p \tag{9.9}$$

S is illustrated on Fig. 9.1. $S = (\Sigma, Q, \Pi, \pi, \delta, \sim_a, \sim_b)$ where

- $\Sigma = \{a, b\}$
- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Pi = \{p\}$
- $\pi(q) = \begin{cases} \{p\} & q \in \{q_1, q_2, q_3\} \\ \emptyset & q = q_4 \end{cases}$
- $\delta(q_1, a) = \delta(q_3, a) = \{\ \{q_2, q_4\}\ \}$
- $\delta(q_1, b) = \{\ \{q_2\}\ \}$
- $\delta(q_3, b) = \{\ \{q_4\}\ \}$
- $\delta(q_2, a) = \delta(q_2, b) = \{\ \{q_2\}\ \}$
- $\delta(q_4, a) = \delta(q_4, b) = \{\ \{q_4\}\ \}$
- $\sim_a = \{(q_1, q_3), (q_3, q_1), (q_1, q_1), (q_2, q_2), (q_3, q_3), (q_4, q_4)\}$
- $\sim_b$ is the identity relation on Q

Clearly, S is an AETS: for every $q \in Q$ if $Q_a \in \delta(q, a)$ and $Q_b \in \delta(q, b)$ then $|Q_a \cap Q_b| = 1$, and $\sim_a$ and $\sim_b$ are equivalence relations. (9.7) holds for any $q \in Q$ (and for both a and b). To see that

$$S, q_1 \models \langle\langle\{a\}\rangle\rangle \bigcirc p \tag{9.10}$$

holds, let $f_a : Q^+ \rightarrow 2^Q$ be a strategy such that $f_a(q_1) = \{q_2, q_4\}$, and let $\lambda \in out(q_1, \{f_a\})$. Clearly, since the only choice available to b in q_1 is $\{q_2\}$, $\{\lambda[1]\} = \{q_2, q_4\} \cap \{q_2\} = \{q_2\}$. $S, \lambda[1] \models p$, so (9.10) holds. Assume that

$$S, q_1 \models K_a\langle\langle\{a\}\rangle\rangle \bigcirc p \tag{9.11}$$

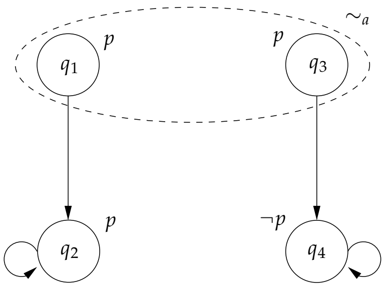

Figure 9.1: The AETS S.

Since $q_1 \sim_a q_3$, $S, q_3 \models \langle\langle\{a\}\rangle\rangle \bigcirc p$. Let f_a be any strategy for a. $f_a(q_3) = \{q_2, q_4\}$ since $\delta(q_3, a)$ is singular. Let $\lambda' \in out(q_3, f_a)$. Since $\delta(q_3, b)$ is singular, $\{\lambda'[1]\} = \{q_2, q_4\} \cap \{q_4\} = \{q_4\}$. Then, $S, \lambda'[1] \models p$ which contradicts the fact that $p \notin \pi(q_4)$. Thus, (9.11) does not hold, which shows (9.9).

As mentioned, not only is (9.7) not a sufficient condition for (9.8); it is also not a necessary condition. I now show this by constructing an AETS S' which is a model for (9.8) for any a, T and ϕ, for which (9.7) does not hold. S' is illustrated on Fig. 9.2. $S' = (\Sigma, Q, \Pi, \pi, \delta', \sim_a, \sim_b)$ where

- $\delta'(q, a) = \begin{cases} \{\ \{q_2\}\ \} & q = q_3 \\ \delta(q, a) & \text{otherwise} \end{cases}$
- $\delta'(q, b) = \begin{cases} \{\ \{q_2\}\ \} & q = q_3 \\ \delta(q, b) & \text{otherwise} \end{cases}$

and $\Sigma, Q, \Pi, \pi, \delta, \sim_a$ and $\sim_b$ is as defined above. Clearly, S' is still an AETS.

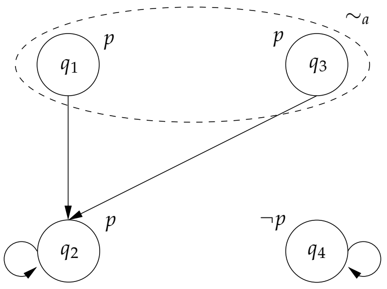

Figure 9.2: The AETS S'.

Let $T \in \{\bigcirc, \Box, F\}$ and ϕ be a formula. To show that $S', q \models \langle\langle\{a'\}\rangle\rangle T\phi \leftrightarrow K_a\langle\langle\{a'\}\rangle\rangle T\phi$ for any $q \in Q$ and $a \in \{a, b\}$, it suffices to show that

$$S', q_1 \models \langle\langle\{a\}\rangle\rangle T\phi \Leftrightarrow S', q_3 \models \langle\langle\{a\}\rangle\rangle T\phi \tag{9.12}$$

since it follows trivially for $a' = a$ and $q \in \{q_2, q_4\}$ and for $a' = b$ for any q. It is easy to see that the only computation starting in q_1 is $\lambda_1 = q_1 q_2 q_2 \cdots$ and the only computation starting in q_3 is $\lambda_2 = q_3 q_2 q_2 \cdots$. Satisfaction of a formula ψ in a state q depends only on i) $\pi(q)$, ii) the states accessible for each agent from q and iii) the set of possible remaining computations starting in q. For q_1

and q_3, i) $\pi(q_1) = \pi(q_3)$, ii) $q_1 \sim_{a'} q'$ iff $q_3 \sim_{a'} q'$ for $a' \in \{a, b\}$ and iii) the set of possible remaining computations starting in q_1 or q_3 is $\{q_2 q_2 \cdots\}$. Thus, $S', \lambda_1[k] \models \phi$ iff $S', \lambda_2[k] \models \phi$ for any k, and (9.12) holds. However, (9.7) does not hold for S': $q_1 \sim_a q_3$ but $\delta'(q_1, a) \neq \delta'(q_3, a)$.

In other words; there are models of the schema (9.8) in which (9.7) do not hold, and there are structures in which (9.7) hold which are not models of the schema (9.8).

One reason that (9.8) may intuitively seem to describe (9.7) is that e.g. $\langle\langle\{a\}\rangle\rangle \bigcirc \phi$ may be interpreted as a statement about agent a's capabilities. This is an imprecise interpretation, as discussed in Section 7.6.2. It may be that $\langle\langle\{a\}\rangle\rangle \bigcirc \phi$ holds because the rest of the system will deterministically make ϕ true in the next state — no matter what agent a does. In other words, it may be that $\langle\langle\emptyset\rangle\rangle \bigcirc \phi$ holds — which trivially implies $\langle\langle\{a\}\rangle\rangle \bigcirc \phi$. For example, (9.10) above holds because the system will deterministically go to state q_2 from q_1. a's capabilities in q_1, described by $\delta(q_1, a)$, is to take the system either to a state where p is true (q_2) or to a state where p is false (q_4), which contrasts with the intuitive interpretation of (9.10) as a statement about a's capabilities.

This inability to syntactically express "local" properties of the agents' capabilities, i.e. about $\delta(q, a)$ in AETSs or about $d_a(a)$ in concurrent game structures, is inherent in ATL. A solution to this problem is *rules and rule operators*. Unlike the temporal operators, the rule operators express "local" properties about the agents' capabilities, e.g. about their mechanisms. I now argue that we can characterize incomplete information better in DSEL by using rule operators. A syntactic property similar to (9.8) with rule operators instead of temporal operators is

$$\Diamond_i T \rightarrow \Big(\bigwedge_{1 \le j \le n} \overset{\leadsto}{\Diamond}_{ij} T_j^R \leftrightarrow \langle\langle\emptyset\rangle\rangle \Box \big(\Diamond_i T \rightarrow \bigwedge_{1 \le j \le n} \overset{\leadsto}{\Diamond}_{ij} T_j^R\big)\Big) \tag{9.13}$$

(9.13) says that if agent i have the capabilities expressed by the rules T_j^R in a state q, then he will have the same capabilities in every future state he cannot discern from q (e.q. where his local epistemic state is the same). Of course, the ATEL property (9.7), in DSEL expressed as

$$s_i = s'_i \Rightarrow d_i(s) = d_i(s') \tag{9.14}$$

with $s = (s_1, \ldots, s_n), s' = (s'_1, \ldots, s'_n)$, already holds by definition. (9.13) is valid (Lemma 7.8).

The next question is, of course, whether (9.14) is a *necessary* condition for (9.13). The answer is "no", and the reason is twofold. First, rule operators can only be used to describe mechanisms which are Cartesian products. Second, (9.13) only describes *possible future states*, not *all* states. If we imagine that the mechanisms were defined as relations over global states rather than as relations between local states and global states, the semantic condition

$$\begin{gathered}\text{If there are } T_j^R \text{ such that } \overrightarrow{R}_\pi, s \models \textstyle\bigwedge_{1 \le j \le n} \overset{\leadsto}{\Diamond}_{ij} T_j^R \\ \text{then for all } s' \text{ reachable from } s : s_i = s'_i \Rightarrow d_i(s) = d_i(s')\end{gathered}$$

where s' is reachable from s iff s' is a state in a computation starting in s, is a necessary condition for (9.13)[11].

In summary; we have looked at syntactic expressions of the property that an agent should be able to do the same thing in indiscernible states. In ATEL, (9.8) was suggested, but this property is not necessarily true even if (9.7) is true. In DSEL we proposed to use (9.13) which is always true when the semantic condition (9.14) is true. This comparison illustrates the fact that rule operators adds expressiveness to the language, because they can express properties of the capabilities of single agents which cannot necessarily be expressed with temporal operators.

9.4.3 Comparison

An obvious observation when comparing DSEL with ATEL, is that if formulae starting with epistemic operators or rule operators are viewed as atomic propositions, DSEL *is* an ATEL logic (with unspecified accessibility relations) since it is an ATL logic[12] and ATEL is a generalization of ATL. Although both logics are integrations of epistemic logics and ATL, the concept of "knowledge" in the two logics are fundamentally different. Incorporating these two concepts into a logic of implicit and explicit knowledge by basing DSEL on ATEL instead of ATL seems to be quite straightforward — the accessibility relations in the structure induced by a mechanism would relate states with the same local state and would describe implicit knowledge with S5 properties.

Of course, a comparison of the two types of "static" knowledge, i.e. knowledge in a fixed state, is exactly the same as a comparison between the semantics of SSEL and Kripke semantics. "Dynamic" knowledge in DSEL is intended to model, in addition to belief revision and communication, "static" knowledge in ATEL (or in Kripke semantics in general). For example, in modal epistemic logic and ATEL,

$$(K_i p \wedge K_i(p \to q)) \to K_i q$$

holds, expressing the fact that the agent's knowledge is closed under modus ponens, while in DSEL

$$(\triangle_i\{p, p \to q\} \wedge \overset{\leadsto}{\triangle}_{ii}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{b\}}\}) \to \langle\langle\{i\}\rangle\rangle \bigcirc \triangle_i\{p\}$$

holds, expressing the fact that if the agent knows $p, p \to q$ *and* modus ponens, then he *could* "close" the formulae $p, p \to q$ under modus ponens if he wanted to.

The concurrent game structures in DSEL are restricted as discussed in the introduction to Section 7.4. The next question is how these restrictions affect the properties of DSEL compared to ATEL. The restriction that the set of possible actions should depend only on the local state was discussed in Section 9.4.2. The restriction to a specific δ gives the properties of communication in

[11]Let $\overrightarrow{R}_\pi, s \models \bigwedge_{1 \le j \le n} \overset{\leadsto}{\Diamond}_{ij} T_j^R$; $R_i(s) = [\![T_1^R]\!](s_i) \times \cdots \times [\![T_n^R]\!](s_i)$. Let s' be reachable from s; $s' = \lambda[k]$ for some computation starting in s. Let $s_i = s'_i$. Let T be such that $[T] = s_i = s'_i$. By (9.13), $\overrightarrow{R}_\pi, s' \models \bigwedge_{1 \le j \le n} \overset{\leadsto}{\Diamond}_{ij} T_j^R$, and $R_i(s') = [\![T_1^R]\!](s'_i) \times \cdots \times [\![T_n^R]\!](s'_i) = R_i(s)$. Thus, $d_i(s) = R_i(s) = R_i(s') = d_i(s')$.

[12]With the slight generalization of e.g. Q to a not necessarily finite set, as described in Section 7.4.1.

Lemma 7.9, which do not hold in general in ATEL. In theory ATEL allows for more general models of communication, but DSEL seem to be general enough for practical purposes. For example, a special agent can act as an "environment" through which communication occur. Of course, the restriction of the state space to a very specific one, i.e. the Cartesian products of possible syntactic epistemic states, restricts the applicability of DSEL to this specific domain as opposed to ATEL, which can be used to model any system to which knowledge can be ascribed. Note that, as a result of the nature of the state space, π, the truth assignment of the primitive propositions, plays a different role in DSEL than in ATEL. In the former, π does not depend on the current state of the system, and have a smaller influence on the semantics of the language.

As discussed in Section 9.4.1 the definition of strategies in ATEL does not take incomplete information into account. The same problem does not occur in DSEL, where the exactly the same restrictions as the ones proposed by Jamroga (2003) for ATEL are imposed on the structures: first, actions are explicit, making it easy to identify the same action in different states, second, the available actions are the same in indiscernible states and third, strategies must map indiscernible states to the same action. A further restriction on strategies in DSEL is that they depend only on the current state and not on the history. Compared to ATEL where knowledge is used as an ascribed notion, this latter restriction is important for DSEL as a theory of explicit, syntactic knowledge.

When rules and rule operators are taken into account, DSEL can no longer be seen as a specialization of ATL (or ATEL). In Section 7.6.2 it was argued that rule operators can be used to express *local* properties of agents' capabilities which temporal operators do not capture in a natural way. One such property, "an agent must be able to perform the same actions in indiscernible states", was discussed in Section 9.4.2. A "natural" expression of this property with temporal operators in ATEL was shown to express not only local capabilities, while a partial expression of the property using rule operators in DSEL was shown to always hold given the semantical restriction. Another property which seem to be easier to express with rule operators is monotonicity of knowledge. As discussed in Section 7.6.1, in DSEL monotonicity can be expressed by the formula

$$\overset{\rightsquigarrow}{\nabla}_{ii}\{\frac{t}{t \sqcup u}\}$$

or by the schema

$$\triangle_i T \to \langle\langle\emptyset\rangle\rangle\Box \triangle_i T$$

In ATEL, without rule operators, a schema must be used, e.g.[13]

$$K_i\phi \to \langle\langle\emptyset\rangle\rangle\Box K_i\phi$$

9.5 Interpreted Multi-agent Systems

Fagin *et al.* (1995) presents, partly based on their previous research, a general framework for ascribing knowledge to multi-agent systems. A multi-agent system can be *any* system composed of interacting agents, and knowledge is used

[13]The notion of monotonicity is of course somewhat different in DSEL and ATEL due to the difference in the two "knowledge" concepts. In the latter monotonicity can not be required to hold for e.g. certain temporal formulae, and ϕ in the schema must be restricted.

as an external notion for analyzing the system rather than a concept describing something the agents can compute, act upon and answer questions about.

Formally, L_i is a set of *local states* for agent i with L_e being the local states for a special agent called the environment, and the set of *global states* is $\mathcal{G} = L_e \times L_1 \cdots \times L_n$. A *run* over $\mathcal{G}$ is a function $r : \mathbb{N} \to \mathcal{G}$, and a *point* is a combination of a run and a number, (r, m). A *system* R over $\mathcal{G}$ is a set of runs over $\mathcal{G}$. An *interpreted system* (R, π) over $\mathcal{G}$ is a system R over $\mathcal{G}$ together with a truth assignment $\pi(s) : \Theta \to \{\mathbf{true}, \mathbf{false}\}$ for each global state $s \in \mathcal{G}$, where Θ is the set of primitive propositions.

By using the assumption that an agent can only discern between points in which his own state differs, an interpreted system $\mathcal{I} = (R, \pi)$ can be seen as inducing a Kripke structure $M_{\mathcal{I}} = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ where S is all points in $\mathcal{I}$ and $((r, m), (r', m')) \in \mathcal{K}_i$ iff $r(m)^i = r'(m')^i$ where the superscript i denotes the ith component. Thus, an interpreted system can be seen as a description of a Kripke structure, in which the language of modal epistemic logic can be interpreted — knowledge, particularly, as *implicit* knowledge. If this language is extended with temporal operators such that if ϕ, ψ are formulae then $\Box\phi, \Diamond\phi, \bigcirc\phi$ and $\phi\mathcal{U}\psi$, meaning "ϕ is true always/eventually/next time/until ψ is true", respectively, the semantics can be easily extended in the usual way: the formulae are true in a point (r, m) in an interpreted system $\mathcal{I}$ iff ϕ is true in (r, m') for all $m' \geq m$, ϕ is true in (r, m') for some $m' \geq m$, ϕ is true in (r, m') for $m' = m + 1$, and there is some $m' \geq m$ such that ψ is true in (r, m') and ϕ is true in (r, m'') for all $m \leq m'' \leq m'$, respectively. In this framework, time is discreet and linear — there is no concept of branching time and no path quantifiers.

Several concepts for describing runs are presented and formalized. For example, a set of *actions* for each agent is used to describe transition functions which maps each agent's choice and the current global state into a new global state, while a *protocol* is a function which maps a local state for an agent into a set of actions. Thus, actions, transition functions and protocols corresponds loosely to ACT, δ, d and strategies in concurrent game structures with incomplete information (see Section 7.4.2).

Restrictions on the framework to describe a *message passing system*, in which the agents can communicate through the environment, is presented. It is not necessary to go into details for this discussion, but local states must include message passing events and the sets of actions must include message passing actions. A version of the Byzantine generals problem, simultaneous Byzantine agreement, is worked out in detail through a careful analysis of the attainment of common knowledge in such systems[14]. For example, it is formally proved that $m + 1$ rounds of communication, where m is the maximal number of traitors, is necessary and sufficient to achieve agreement about the initial value.

9.5.1 Comparison

There are at least two aspects of the work on interpreted multi-agent systems in (Fagin *et al.*, 1995) which are interesting in comparison to the work in this

[14]Only certain types of traitor behaviour, called *benign failures* (crash failures, sending-omission failures, general-omission failures), are considered. In Chapter 8, no such restrictions were made and the traitors could actively try to deceive the loyal generals. This latter type of behaviour is called *Byzantine failures*.

thesis. First, the framework can be compared to the framework for the dynamic logic developed in Chapter 7, and, second, the application mentioned above can be compared to the Byzantine generals example in Chapter 8.

The framework is effectively a tool for describing classes of Kripke structures which can change with time. The notion of knowledge is the same as in modal epistemic logic, except from the fact that it can change with time. The only difference from the language of modal epistemic logic is the addition of temporal operators. Unlike the temporal operators in ATL, however, these model a concept of linear rather than branching time. Concepts such as actions and protocols are not directly a part of the semantics of the language; rather, they are used to describe classes of systems of interest, i.e. "legal" runs. As noted above, in frameworks derived from ATL, such as Alternating time Temporal Epistemic Logic (ATEL, discussed in the next section) and the one presented in Chapter 7, the concepts of actions and protocols are formalized as part of the semantical structures. In addition, ATL uses a more general model of time (branching versus linear), and in the frameworks just mentioned the more powerful temporal language can sometimes be used instead of meta-logical description of model classes. One interesting part of the description of the framework is the development of message passing systems. Such systems can be modeled in the framework from Chapter 7 in a similar manner, with computations corresponding to runs and a particular agent "being" the environment. For example, while in all the examples in this thesis the agents have communicated directly to each other by "inserting" formulae into epistemic states, unreliable communication can be modeled by sending messages through the environment agent. Since an advanced model of message passing was not needed for the worked example (Byzantine generals, Ch. 8) due to the fact that communication there was assumed to be direct and reliable, such applications are left for future work.

The application in the analysis of simultaneous Byzantine agreement shows some very important correspondences between common knowledge, agreement and properties of message passing systems. The application is mainly on a meta-logical level, reasoning about classes of runs, since many of the developed concepts cannot be expressed in the logical language. The example in Chapter 8 demonstrates the use of several language constructs not found in the interpreted systems language, such as

- The use of $\triangledown_i \emptyset$ to denote the fact that agent i knows *nothing* in the initial situation.
- The use of the temporal connectives $\langle\langle\emptyset\rangle\rangle\Box$ and $\langle\langle G\rangle\rangle\mathcal{F}$.
- The use of $\wedge_{ij} \overset{\rightsquigarrow}{\triangle}_{ij}\, T^{ij}$ to denote the fact that the generals must use the protocols T^{ij}.
- The restriction of the possible messages that can be sent ("oral messages"), by the expression $\bigwedge_{i,j} \overset{\rightsquigarrow}{\triangledown}_{ij}\{\frac{t}{\triangle_j\{\neg\neg\triangle_i\{a\}\}}, \frac{t}{\triangle_j\{\neg\triangle_i\{a\}\}}, \frac{t}{\emptyset}\}$

There is some overlap between the results shown in the two applications — Fagin *et al.* (1995) do not show the negative result (Section 8.3), however. Extending the logic in Chapter 7 with implicit knowledge and adapt the results involving common knowledge could be interesting for future work.

9.6 Dynamic Epistemic Logic

The motivation behind *dynamic epistemic logic* (Duc, 1997b) is very similar to the one presented in this thesis. Duc argues that many intuitions about knowledge is lost when the LOP is attacked by weakening standard epistemic logic, and proposes a solution to this problem. The main idea is that, in general, we cannot assume that the knowledge of an agent is closed under *any* logical law. That is, the agents described by the traditional logics does not exist. Furthermore the existence of logics that describe real agents is questioned, because real agents do not tend to reason in a fixed pattern. A problem with many of the proposed solutions to the LOP is that they sacrifice rationality for non-omniscience; this is a dilemma between logical ignorance and logical non-omniscience.

The proposed solution is to *dynamize* epistemic logic. The idea is that an agent knows a new fact if it follows from the agent's other knowledge *and* the agent performs an inference action. Such an action is modeled as the selection and use of an inference rule or an axiom. Dynamic logic (see e.g. Harel (1984)) is used to formalize actions. For example, if R is a inference rule, then intention of the formula $[R_i]K_i\phi$ is that agent i will always know ϕ after using rule R. However, to avoid fixing an axiomatization for each agent and to avoid a very complex system, the individual actions for each agent are replaced by an auxiliary action F_i, denoting an arbitrary (but including at least one action) and non-deterministic sequence of actions. The intended meaning of $\langle F_i\rangle\phi$ is that ϕ is true after some train of thought of agent i. The dual operator $[F_i]$ is also used; $[F_i]\phi \equiv \neg\langle F_i\rangle\neg\phi$, meaning that ϕ is true after every train of though of agent i.

Formally, *dynamic epistemic logic* for n agents is defined as follows. The language L_{DE} is defined as a superset of $\mathcal{L}_n(\Phi)$ (p. 9):

$$\begin{gathered}\mathcal{L}_n(\Phi) \subseteq L_{DE}\\ \text{If } \phi \in L_{DE}\text{, then } \neg\phi \in L_{DE}\\ \text{If } \phi, \psi \in L_{DE}\text{, then } (\phi \rightarrow \psi) \in L_{DE}\\ \text{If } \phi \in L_{DE}\text{, then } \langle F_i\rangle\phi \in L_{DE}\end{gathered}$$

In addition the sublanguage $L_E^+ \subseteq \mathcal{L}_n(\Phi)$ is used below. L_E^+ contains all the formulae of $\mathcal{L}_n(\Phi)$ without any occurrences of the knowledge operators K_i and is closed under the following conditions:

$$\begin{gathered}\text{If } \phi, \psi \in L_E^+\text{, then } (\phi \wedge \psi) \in L_E^+\\ \text{If } \phi, \psi \in L_E^+\text{, then } (\phi \vee \psi) \in L_E^+\\ \text{If } \phi \in L_E^+\text{, then } K_i\phi \in L_E^+\end{gathered}$$

A "dynamic" version of *S4n*, the logical system $DES4_n$, has the following axiom schemata:

$$\phi \to (\psi \to \phi) \tag{9.15}$$
$$(\phi \to (\psi \to \gamma)) \to ((\phi \to \psi) \to (\phi \to \gamma)) \tag{9.16}$$
$$(\neg\psi \to \neg\phi) \to (\phi \to \psi) \tag{9.17}$$
$$[F_i](\phi \to \psi) \to ([F_i]\phi \to [F_i]\psi) \tag{9.18}$$
$$[F_i]\phi \to [F_i][F_i]\phi \tag{9.19}$$
$$K_i\phi \wedge K_i(\phi \to \psi) \to \langle F_i\rangle K_i\psi \tag{9.20}$$
$$K_i\phi \to \phi \tag{9.21}$$
$$K_i\phi \to [F_i]K_i\phi, \text{ if } \phi \in L_E^+ \tag{9.22}$$
$$\langle F_i\rangle K_i(\phi \to (\psi \to \phi)) \tag{9.23}$$
$$\langle F_i\rangle K_i((\phi \to (\psi \to \gamma)) \to ((\phi \to \psi) \to (\phi \to \gamma))) \tag{9.24}$$
$$\langle F_i\rangle K_i((\neg\psi \to \neg\phi) \to (\phi \to \psi)) \tag{9.25}$$
$$\langle F_i\rangle K_i(K_i\phi \to \phi) \tag{9.26}$$
$$K_i\phi \to \langle F_i\rangle K_iK_i\phi, \text{ if } \phi \in L_E^+ \tag{9.27}$$

and the following inference rules:

$$\frac{\vdash_{DES4_n} \phi, \vdash_{DES4_n} \phi \to \psi}{\vdash_{DES4_n} \psi} \tag{9.28}$$
$$\frac{\vdash_{DES4_n} \phi}{\vdash_{DES4_n} [F_i]\phi} \tag{9.29}$$

The use of L_E^+ avoids certain potential problems with monotonicity of knowledge.

Dynamic epistemic logic captures the idea that an agent can get to know any logical consequence of its knowledge if it thinks hard enough. The agents in this approach are not logically omniscient. For example, as shown in the paper, the knowledge necessitation rule (p. 10)

$$\frac{\vdash_S \phi}{\vdash_S K_i\phi}, 1 \leq i \leq n \qquad \textbf{Nec}$$

is not derivable in $DES4_n$. The following rule is, however:

$$\frac{\vdash_{DES4_n} \phi}{\vdash_{DES4_n} \langle F_i\rangle K_i\phi}$$

This rule says that an agent *can* get to know any theorem. Thus agents are both logically non-omniscient and non-ignorant. Also, agents are rational; the approach does not attribute non-omniscience to failure to follow the laws of logic. Other examples of non-ignorance is that the following (where ϕ and ψ are $\mathcal{L}_n(\Phi)$-formulae without any occurrences of the knowledge operators) are theorems of $DES4_n$:

$$K_i(\phi \wedge \psi) \to \langle F_i\rangle K_i\phi$$
$$(K_i\phi \wedge K_i\psi) \to \langle F_i\rangle K_i(\phi \wedge \psi)$$

9.6.1 Comparison

Duc (1997b) does not define a semantics for dynamic epistemic logic[15].

Let us look at the single agent case first. In this case, it seems that dynamic epistemic logic can be expressed in DSEL. Taking

$$\langle F_i \rangle \phi \equiv \langle\langle \{i\} \rangle\rangle \mathcal{F} \phi$$

and

$$[F_i]\phi \equiv [\![\{i\}]\!] \Box \phi$$

seems to capture the intended semantics. Under this definition, axioms (9.18) and (9.19) are valid, and inference rule (9.29) preserves validity, in DSEL. Axioms (9.15)–(9.17) are of course also valid, and (9.28) preserves validity. (9.21) and (9.22) are axiomatizations of truth of knowledge and monotonicity and can be expressed in DSEL by

$$\triangle_i\{\alpha\} \to \alpha$$

and

$$\triangle_i\{\alpha\} \to [\![\{i\}]\!] \triangle_i \{\alpha\}$$

respectively. The rest of the axioms is an axiomatization of the reasoning mechanism, and can be expressed in DSEL either as above or as rules.

Thus, $DES4_n$ can be seen as an axiomatization of the two temporal operators along with a resoning mechanism for the agents.

For the multi-agent case, the correspondance to DSEL, and ATL, is sligthtly different because the notion of time in dynamic epistemic logic is *subjective* with respect to each agent — there is no axiomatization of a common clock in $DES4_n$. However, since there is no axiomatization the properties of interaction between agents at all, or mentions of such properties in the paper, it seems that $DES4_n$ is not primarily intended as a multi-agent logic.

9.7 Active Logics

Active Logics (Elgot-Drapkin *et al.*, 1999)[16] is a framework, based on first-order logic[17], for "reasoning situated in time". The idea is that agents' reasoning progress in discrete steps, and that an agent should be able to reason about what agents (including itself) do or do not know after a certain number of steps.

The idea is partly motivated by the role of time in commonsense reasoning. Main points are that, first, the reasoning of an agent seldom has an endpoint but goes on indefinitely and, second, that agents are fallible – they can believe

[15]Duc (1995) defines a semantics, based on temporal frames, for a similar, but less expressive logic.

[16]Formerly (Drapkin & Perlis, 1986; Elgot-Drapkin, 1988; Elgot-Drapkin & Perlis, 1990) known as *step logics*.

[17]Nirkhe, Kraus, & Perlis (1994) present a modal logic analog to a particular SL_5 (see below) step logic. They provide a semantics based on a combination of the Montague-Scott approach and *timelines*, and a sound and complete axiomatization with respect to this semantics. The LOP is only partly solved, because Montague-Scott semantics does not discriminate between formulae with the same intentions (e.g. all tautologies) (see Section 9.2). The authors comment that this problem can be solved by introducing a syntactic component similar to an awareness operator.

in contradictions without believing in "everything" (as they would if they were logically omniscient — the authors call this "swamping"). Reasoning should therefore, they argue, not be modeled only as an ultimate set of conclusions.

Semantically, the active logic approach views agents as having a *finite belief set* and an *inference engine*.

Active logics is a unified logical framework, rather than a single logic. I will not go into all the extensive details here, but discuss some of the aspects. Within the framework, eight different active logic pairs are defined, corresponding to different agent languages and logical abilities of agents such as keeping track of time, self knowledge and contradiction handling. For version n, SL_n is the logic an agent reasons in and SL^n is a (consistent) first-order meta-theory about the agent. The meta-theory includes a binary predicate K; $K(i, \phi)$ denotes the fact that ϕ is known at time step i (technically, formulae which can be known can be represented as a term). Some of the active logics, e.g. $n = 7$, includes a similar predicate in the local agent theory. An active logic is characterized by the following three parts:

1. A first-order language $\mathcal{W}$. This is the internal language the agents reason in.

2. An *observation function* $OBS : \mathbb{N} \rightarrow \wp^{fin}(\mathcal{W})$, giving a finite set of observations for each time step. An observation is a wff the agent gets to know at the given time step.

3. An *inference-function* $INF : \mathcal{H} \rightarrow \wp^{fin}(\mathcal{W})$ mapping a history, where $\mathcal{H}$ consists of all chains of pairs of belief sets and observation sets, to a finite set of wffs. Intuitively, the inference-function gives the set of beliefs for the next step (which do not necessarily include those of the previous step). An inference-function can be represented as a set of inference rules, with the meaning that the function returns the results of applying all the rules to all the wffs from the last step in the history.

In (Elgot-Drapkin & Perlis, 1990), the active logic pair $\langle SL_7, SL^7 \rangle$ is used to demonstrate, among other tings, default reasoning and contradiction detection and handling. To this end, the internal logic is non-monotonic. Each agent can keep track of time by using a predicate $Now(i)$ and a rule replacing $Now(i)$ with $Now(i + 1)$ (an example of non-monotonicity). This rule illustrates the idea that "reasoning is situated in time", that reasoning itself takes time — in english: "if now the time is i then now the time is $i + 1$". A subclass of the formulae is time-stamped. Note that an agent can know at time i the formula $K(j, \alpha)$; i.e. that he knew (if $j \leq i$) α at time j. A formula is inherited from the previous step only if it does not form a contradiction with another believed formula at that step. Negative introspection is modeled (in an agent's local reasoning) by including $\neg\phi$ at step $i + 1$ if ϕ is not believed at step i and ϕ is a closed subformula of a formula believed at step i (this last condition can be seen as a form of awareness).

The active logics framework also provide models for a range of other aspects of reasoning. For example, bounded memory is modeled by using a more advanced model of memory than belief sets, involving a short term and a long term memory.

9.7.1 Comparison

Although many of the motivations behind the active logics approach are similar to those presented in this thesis, the approaches are quite different. Active logics provides richer and more flexible framework which is more applicable to "real" situations; this is partly due to the fact that it is based on first order logic. SSEL and DSEL, on the other hand, are designed to model an isolated aspect of knowledge and reasoning — finite syntactic epistemic states.

The similarities of the two appraches includes the fact that both model agents as having a storage of formulae and a reasoning mechanism. Furthermore, the storage is required to be finite in both approaches. The storage can in principle contain arbitrary formulae also in active logics, although restrictions are placed on inheritance from time point to time point in practice. Both approaches can avoid "swamping", and the contradiction handling mechanism in active logics can also be implemented in DSEL. Since active logics are based on first order logic and formulae can be expressed as terms, it seems that the operators $\triangledown_a$ and $\overset{\rightsquigarrow}{\triangledown}_{aa}$ can be expressed by using variables — for example $x \neq p \rightarrow \neg K(i, x)$ for $\triangledown_a\{p\}$.

A main difference between the two approaches is the model of time. In the words of (Elgot-Drapkin *et al.*, 1999), "the focus of active logic is not primarily to be able to reason *about* time, but rather to be able to reason *in* time". The main use of time in active logics is to model the fact that reasoning takes time. In active logics the agents use the "whole" mechanism, all the rules if inference rules are used to model the mechanism, at each time step. Thus there is no concept of different possibilities, and time is modeled as being linear instead of branching. As a consequence of this, there is no concept of CTL or ATL - type temporal connectives, strategies or cooperation, like in DSEL. In fact, in one sense an active logic is a single agent logic. Although agents can reason about other agents, there is no formal multi agent model.

Note that the concept of *rules* is not formalized in active logics as a part of the agent language — rules are used as a meta-logical descriptions of the inference function — but can be easily expressed in the first order language. For example, in an analysis of the three wise men puzzle using a SL_5 step-logic[18], Elgot-Drapkin (1991b) includes the formula

$$(\forall j)K_2(j, (\forall i)(\forall x)(\forall y)[K_3(i, x \rightarrow y) \rightarrow (K_3(i, x) \rightarrow K_3(s(i), y))])$$

in $OBS(i)$, to model the fact that the agent knows that wise man 2 knows at every step that wise man 3 uses the rule modus ponens. As already mentioned; the semantics of "knowing a rule" is different than in DSEL, since the agents always use all the rules in each step in active logics.

In summary; the motivation behind active logics and DSEL is similar, the former has much higher expressive power but a fixed deterministic mechanism while the latter have non-deterministic control mechanisms and is based on a model of branching time.

[18]See also (Elgot-Drapkin, 1991a) for a version of the puzzle with two wise men only.

Chapter 10

Conclusions and Future Work

10.1 Summary

The problem investigated in this thesis is the modeling of the explicit knowledge of deliberative reasoners who represent their knowledge syntactically, in a logical framework. The problem with using traditional epistemic logic based on modal logic to this end, is that they describe agents who know all the infinitely many consequences of their knowledge. The reason for this problem is that in these logics there is no distinction between knowledge and reasoning — knowledge is modeled as everything which can be obtained by (sound and complete) reasoning. The model in this thesis solves this problem by modeling knowledge and reasoning as two different concepts: Part II presents a theory of static finite syntactic epistemic states, and Part III presents a theory of the dynamics of such states, i.e. how they change over time as a result of reasoning and/or communication.

10.1.1 Part II

In Part II the logic SSEL (Static Syntactic Epistemic Logic) is developed. The semantic assumptions about the agents includes that their epistemic states are sets of "formulae" in an *object language*, but include no restrictions on closure or consistency of such sets. The object language is a propositional language parameterized by the number of agents n and primitive propositions Θ, and includes two epistemic operators: $\triangle_i X$ expressing the fact that agent i knows at least the set X and $\bigtriangledown_i X$ expressing the fact that agent i knows at most the set X. Thus, it is assumed that the agents can represent sets. A meta language EL for reasoning about such agents is introduced. The meta language is similar to the object language, but uses terms for expressing sets of object language formulae, and has primitives for expressing relations between sets. A sound and complete calculus is developed for this latter part of the language (the "term language"). One of the main goals in this thesis was to model agents with *finite* epistemic states. To this end a more general model where the agents are not restricted to finite states was useful as an intermediate result. Semantics for this case was presented, where each agent's epistemic state is represented by a set of object language formulae — possibly in addition to a particular formula $*$ which does not exist in the meta language. It is argued that the agents must be able to know a formula like $*$ not expressible in the meta language in order for there to exist an axiomatization (with finite inference rules) of the semantics. The structures representing each agent as a (possibly infinite) subset of the object language, extended with $*$, are called *General Knowledge Set Structures (GKSSs)*. A logical system *EC* for the language *EL* is developed, and proved sound and strongly complete with respect to the set of all GKSSs.

Agents restricted to finite epistemic states, without the $*$ element, are modeled by *Knowledge Set Structures (KSSs)*. This has consequences for (strong) completeness; an infinite set of formulae could e.g. describe an agent with an infinite state. Furthermore, this type of incompleteness is fundamental to the semantics since inconsistency of such descriptions of infinite states cannot always be axiomatized (with finite inference rules). I describe the theories for which the logic *EC* (the same logic as in the unrestricted case) is complete: *finitary theories*. A proof of finitaryness of the empty set, and thus weak completeness of the logic, is presented. The proof uses the fact that *EC* is complete with respect to GKSSs. Although the result seems intuitive, there does not seem to exist a trivial proof of completeness.

Different variations of modal epistemic logic can be obtained by adding axioms corresponding to epistemic properties. In Chapter 6 such extensions of *EC* were investigated. Epistemic axiom schemata, i.e. axioms describing purely epistemic properties, corresponds to removing illegal epistemic states. *EC* extended with a finitary epistemic axiom schema is sound and (weakly) complete with respect to the set of KSSs built from only legal epistemic states. Algebraic conditions on the sets of legal epistemic states induced by an epistemic axiom schema are developed, which are sufficient for the axiom schema to be finitary. Thus, the logic can potentially be extended with an epistemic axiom schema by i) constructing the sets of legal epistemic states, ii) using the algebraic conditions to show that the axiom schema is finitary and iii) *EC* extended with the axiom schema is thus sound and (weakly) complete w.r.t. the KSSs constructed from the legal epistemic states.

10.1.2 Part III

Change in finite epistemic states can be seen as agents moving from point to point in the lattice of such states. In the logic DSEL (Dynamic Syntactic Epistemic Logic) developed in Part III agents still have the same type of epistemic states as in Part II, i.e. finite subsets of the object language, but in addition have *mechanisms*. A mechanism models both reasoning and communication; it maps an epistemic state to a set of n-tuples of sets of object formulae. Each n-tuple represents the information the agent can send to the other agents and to himself before the next time step.

The meta language *EL* from Part II is extended to a meta language *TEL* for reasoning about syntactic epistemic states over time. A key point of the model is that it is non-deterministic; time is branching and agents can cooperate to achieve goals. Therefore, the language is chosen as an extension of the language of ATL, and includes temporal operators such as $\langle\langle\{i,j\}\rangle\rangle\mathcal{F} \triangle_i \{p\}$ which means that agents i and j can cooperate to make i know p in the future. In addition, *rules* and *rule operators* are introduced. Like knowledge operators there are two rule operators for knowing at least and at most a set of rules, respectively. An example of the former is $\overset{\rightsquigarrow}{\triangle}_{11}\{\frac{t\sqcup\{a,a\to b\}}{t\sqcup\{a,a\to b,b\}}\}$ (agent 1 can reason with modus ponens); an example of the latter is $\overset{\rightsquigarrow}{\triangledown}_{22}\{\frac{t}{t\sqcup u}\}$ (every rule agent 2 knows extends the current epistemic state; the mechanism is monotone). Properties of the communication part of the mechanism can be expressed in a similar manner, i.e. $\overset{\rightsquigarrow}{\triangledown}_{12}\{\frac{\{a\}}{\{a\}}\}$ (agent 1 can only tell agent 2 things he (agent 1) knows).

DSEL is an extension of SSEL in the following way: the language *TEL* is an extension of the language *EL*, and given a mechanism and a local state for each agent, DSEL describes a KSS for *each point in time*.

Mechanisms can be seen as inducing ATL-type concurrent game structures, and satisfaction is thus defined as ATL-satisfaction with *TEL* seen as an ATL language. The induced concurrent game structures have certain properties such as incomplete information and non-perfect recall.

The small modus ponens example on p. 99 illustrated cooperation, reasoning and communication in DSEL. Larger examples of more isolated aspects of DSEL was illustrated in the three wise men example (Sec. 7.7.1) (reasoning), and aspects of the Byzantine Generals problem (Ch. 8) (knowing "at most" a set of rules, communication).

10.2 Conclusions

In this thesis a logic of finite syntactic epistemic states is presented.

The static part of the logic, SSEL, extends the theory of traditional syntactic structures in two ways: by requiring epistemic states to be finite and by an operator expressing that *at most* a set of formulae can be known. A consequence is that complete axiomatization is non-trivial, and the thesis presents a theory which is used to show completeness both of a basic system and of the basic system extended with epistemic properties.

The dynamic part of the logic extends the static part and is based on alternating-time temporal logic and models agents with mechanisms for reasoning and communication. The temporal language is extended with rule operators, which can be used to express properties of the mechanisms. Particularly, rule operators can express "local" properties not easily expressed with temporal operators. In ATEL, another epistemic logic based on ATL, there is a problem with the semantics of available choices in indiscernible states, and a proposal of a syntactical characterization does not express the desired semantics. In DSEL this semantical property is inherent, and can be (partly) expressed syntactically by using rule operators instead of temporal operators.

The logic is a solution to the logical omniscience problem without introducing the problem of logical ignorance. It does not follow, in general, from the fact that an agent knows something that he must know something else, but if given proper deduction rules it follows that he can get to know it if he chooses to.

Although DSEL is an extension of SSEL, the latter is a interesting logic in itself. The two frameworks are quite different; SSEL is a description of restricted and very well defined concept and is used to show several theoretical properties about that concept, while DSEL is a more general model which can be applied to several types of problems, but is more complex and does not have as well understood theoretical properties. The relationship between the logics has not been properly investigated yet.

10.3 Future Work

The most obvious future work on the logic SSEL from Part II is further development of the identification of finitary theories. For the case of epistemic axioms, the presented algebraic conditions are sufficient but not necessary. Tighter conditions would be interesting. Deciding finitaryness of general, not necessarily epistemic, axioms should also be investigated. Especially interesting is the $\mathbf{T_i}$ axiom. A general alternative to algebraic conditions of finitaryness could be a syntactic characterization of the form of axiom schemata, i.e. a definition of "the language of finitary schemata".

The logic DSEL from Part III is where most future work is needed. Some areas are:

- The development of a logical calculus. This would involve extending the calculus *EC* for SSEL with an axiomatization of rule- and temporal operators. However, to my knowledge there does not at the time of writing exist a complete axiomatization of ATL. Goranko (2001) provides a partial axiomatization, and van der Hoek & Wooldridge (2003) also discuss properties of the temporal operators in ATL. The properties of rule operators discussed in Section 7.6.2, Lemmas 7.8 and 7.9, could be a part of an axiomatization.

- The relation between DSEL and SSEL should be investigated in greater detail. To this end, an axiomatization (the previous point) could be a great help. For example, finitary epistemic axioms correspond to removing certain illegal epistemic states *now*, while rules and temporals "remove" certain illegal states in the *future*.

- Examples of equipping agents with rules implementing particular mechanisms, e.g. sound and complete propositional reasoning.

- As a generalization of the previous point: equipping agents with mechanisms implementing sound and complete reasoning in the whole object language; i.e. implementing the *EC* calculus.

- Integrate DSEL with ATEL or, equivalently, base DSEL on ATEL instead of ATL, in order to create a logic of both explicit and implicit knowledge as discussed in Section 9.4.3.

- Model message passing systems in DSEL, as discussed in Section 9.5.1.

- Introduce a "syntactic" common knowledge operator.

- Depending on several of the above points (message passing systems, common knowledge): the results about the Byzantine Generals problem and common knowledge discussed in Section 9.5 could be adapted or compared to an extended version of DSEL.

- Extending the object language to mirror *TEL*, in order to model agents who can reason about the reasoning of other agents. This would e.g. allow nested rules and temporals nested inside rules. For example, such a logic could express the fact that agent i knows the following rule: if agent j knows $p, p \rightarrow q$ and agent j knows modus ponens then agent j might know q in the next time step. This is complicated, because the meaning of temporal formulae change by the act of performing computations. Active Logics (Section 9.7) provides a framework modeling this type of reasoning.

Questions of complexity need to be investigated for both SSEL and DSEL.

Bibliography

Alur, R.; Henzinger, T. A.; and Kupferman, O. 1997. Alternating-time temporal logic. In *38th Annual Symposium on Foundations of Computer Science*, 100–109. Miami Beach, Florida: IEEE.

Alur, R.; Henzinger, T.; and Kupferman, O. 1999. Alternating-time temporal logic. In *Compositionality: The Significant Difference*, Lecture Notes in Computer Science 1536. Springer-Verlag. 23–60.

Alur, R.; Henzinger, T.; and Kupferman, O. 2002. Alternating-time temporal logic. *Journal of the ACM* 49:672–713.

Anderson, A. R., and Belnap, N. D. J. 1975. *Entailment, the Logic of Relevance and Necessity*. Princeton: Princeton University Press.

Chomsky, N. 1982. *The Generative Enterprise*. Dordrecht: Foris.

Cresswell, M. J. 1970. Classical intensional logics. *Theoria* 36:347–372.

Dennett, D. C. 1987. *The intentional stance*. The MIT Press, Massachusetts.

Drapkin, J., and Perlis, D. 1986. Step-logics: An alternative approach to limited reasoning. In *Proceedings of the European Conference on Artificial Intelligence*, 160–163.

Duc, H. N. 1995. Logical omniscience vs. logical ignorance. on a dilemma of epistemic logic. In Pereira, C. P., and Mamede, N., eds., *Progress in Artificial Intelligence. Proceedings of EPIA'95*, volume 990 of *LNAI*. Heidelberg: Springer.

Duc, H. N. 1997a. On the epistemic foundations of agent theories. In Singh, M. P.; Rao, A.; and Wooldridge, M. J., eds., *Intelligent Agents IV: Agent Theories, Architectures, and Languages: 4th International Workshop, ATAL'97, Proceedings*, volume 1365 of *LNAI*, 275 – 279. Berlin: Springer.

Duc, H. N. 1997b. Reasoning about rational, but not logically omniscient, agents. *Journal of Logic and Computation* 7(5):633–648.

Eberle, R. A. 1974. A logic of believing, knowing and inferring. *Synthese* 26:356–382.

Elgot-Drapkin, J. J., and Perlis, D. 1990. Reasoning situation in time I: Basic concepts. *Journal of Experimental and Theoretical Artificial Intelligence* 2:75–98.

Elgot-Drapkin, J.; Kraus, S.; Miller, M.; Nirkhe, M.; and Perlis, D. 1999. Active logics: A unified formal approach to episodic reasoning. Technical Report CS-TR-4072.

Elgot-Drapkin, J. J. 1988. *Step-logic: Reasoning Situated in Time*. Ph.D. Dissertation, Department of Computer Science, University of Maryland, College Park, Maryland.

Elgot-Drapkin, J. J. 1991a. A real-time solution to the wise-men problem. In *Proceedings of the AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, 33–40.

Elgot-Drapkin, J. J. 1991b. Step-logic and the three-wise-men problem. In Dean, T., and McKeown, K., eds., *Proceedings of the Ninth National Conference on Artificial Intelligence*, 412–417. Menlo Park, California: American Association for Artificial Intelligence.

Fagin, R., and Halpern, J. Y. 1985. Belief, awareness and limited reasoning. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 491–501.

Fagin, R., and Halpern, J. Y. 1988. Belief, awareness and limited reasoning. *Artificial Intelligence* 34:39–76. A preliminary version appeared in Fagin & Halpern (1985).

Fagin, R.; Halpern, J. Y.; Moses, Y.; and Vardi, M. Y. 1995. *Reasoning About Knowledge*. Cambridge, Massachusetts: The MIT Press.

Fagin, R.; Halpern, J. Y.; and Vardi, M. Y. 1991. A model-theoretic analysis of knowledge. *Journal of the ACM* 38(2):382–428. A preliminary version appeared in *Proc. 25th IEEE Symposium on Foundations of Computer Science*, 1984.

Fagin, R.; Halpern, J. Y.; and Vardi, M. Y. 1996. A nonstandard approach to the logical omniscience problem. *Artificial Intelligence* 79(2):203–240. A preliminary version appeared in Parikh (1990).

Goranko, V. 2001. Coalition games and alternating temporal logics. In *Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, 259–272. Morgan Kaufmann.

Hadley, R. F. 1986. Fagin and halpern on logical omniscience: A critique with an alternative. In *Proc. Sixth Canadian Conference on Artificial Intelligence*, 49 – 56. Montreal: University of Quebec Press.

Hadley, R. F. 1988. Logical omniscience, semantics and models of belief. *Computational Intelligence* 4:17–30.

Halpern, J. Y., and Lakemeyer, G. 1995. Levesque's axiomatization of only knowing is incomplete. *Artificial Intelligence* 74(2):381–387.

Halpern, J. Y., and Lakemeyer, G. 1996. Multi-agent only knowing. In Shoham, Y., ed., *Theoretical Aspects of Rationality and Knowledge: Proceedings of the Sixth Conference (TARK 1996)*. San Francisco: Morgan Kaufmann. 251–265.

Halpern, J. Y., and Moses, Y. 1985. Towards a theory of knowledge and ignorance. In Apt, K. R., ed., *Logics and Models of Concurrent Systems*. Berlin: Springer-Verlag. 459–476.

Halpern, J. Y., and Moses, Y. 1990. Knowledge and common knowledge in a distributed environment. *Journal of the ACM* 37(3):549–587.

Halpern, J. Y.; Moses, Y.; and Vardi, M. Y. 1994. Algorithmic knowledge. In Fagin, R., ed., *Theoretical Aspects of Reasoning about Knowledge: Proc. Fifth Conference*, 255–266. San Francisco: Morgan Kaufmann.

Halpern, J. Y. 1997. A theory of knowledge and ignorance for many agents. *Journal of Logic and Computation* 7(1):79–108.

Harel, D. 1984. Dynamic logic. In Gabbay, D., and Guenthner, F., eds., *Handbook of Philosophical Logic, Volume II: Extensions of Classical Logic*, volume 165 of *Synthese Library*. Dordrecht: D. Reidel Publishing Co. chapter II.10, 497–604.

Heyting, A. 1956. *Intuitionism: An Introduction*. Amsterdam: North-Holland.

Hintikka, J. 1962. *Knowledge and Belief*. Ithaca, New York: Cornell University Press.

Hintikka, J. 1975. Impossible possible worlds vindicated. *Journal of Philosophical Logic* 4:475–484.

Jamroga, W. 2003. Some remarks on alternating temporal epistemic logic. In *FAMAS'03 – Formal Approaches to Multi-Agent Systems, Proceedings*, 133–140.

Konolige, K. 1982. Circumscriptive ignorance. In Waltz, D., ed., *Proceedings of the National Conference on Artificial Intelligence*, 202–204. Pittsburgh, PA: AAAI Press.

Konolige, K. 1984. *A Deduction Model of Belief and its Logics*. Ph.D. Dissertation, Stanford University.

Konolige, K. 1985. Belief and incompleteness. In Hobbs, J. R., and Moore, R. C., eds., *Formal Theories of the Commonsense World*. New Jersey: Ablex Publishing Corporation. chapter 10, 359 – 403.

Konolige, K. 1986a. *A Deduction Model of Belief*. Los Altos, California: Morgan Kaufmann Publishers.

Konolige, K. 1986b. What awareness isn't: A sentential view of implicit and explicit belief. In Halpern, J. Y., ed., *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the First Conference*, 241–250. Los Altos, California: Morgan Kaufmann Publishers, Inc.

Lakemeyer, G. 1987. Tractable meta-reasoning in propositional logics of belief. In McDermott, J., ed., *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI '87)*, 401–408. Milan, Italy: Morgan Kaufmann.

Lakemeyer, G. 1996. Limited reasoning in first-order knowledge bases with full introspection. *Artif. Intell.* 84(1-2):209–255.

Lamport, L.; Shostak, R.; and Pease, M. 1982. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems* 4(3):382–401.

Levesque, H. J. 1984a. A logic of implicit and explicit belief. In *Proceedings of the Fourth National Conference on Artificial Intelligence (AAAI'84)*, 198–202. American Association for Artificial Intelligence.

Levesque, H. J. 1984b. A logic of implicit and explicit belief. Technical Report 32, Fairchild Laboratory of Artificial Intelligence, Palo Alto, US.

Levesque, H. J. 1990. All I know: a study in autoepistemic logic. *Artificial Intelligence* 42:263–309.

McCarthy, J. 1978. Formalization of two puzzles involving knowledge. Unpublished manuscript, Computer Science Dept., Stanford University.

Montague, R. 1968. Pragmatics. In Klibansky, R., ed., *Contemporary Philosophy: A Survey. I.* Florence: La Nuova Italia Editrice. 102–122. Reprinted in (Montague, 1974, pp. 95 – 118).

Montague, R. 1970. Universal grammar. *Theoria* 36:373–398. Reprinted in (Montague, 1974, pp. 222 – 246).

Montague, R. 1974. *Formal Philosophy*. New Haven, CT: Yale University Press.

Moore, R. C., and Hendrix, G. 1979. Computational models of beliefs and the semantics of belief sentences. Technical Note 187, SRI International, Menlo Park, CA.

Moore, R. C. 1983. Semantical considerations on nonmonotonic logic. In Bundy, A., ed., *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, 272–279. Karlsruhe, FRG: William Kaufmann.

Moore, R. C. 1986. Reasoning about knowledge in artificial intelligence. In Halpern, J. Y., ed., *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the First Conference*, 81 – 82. Los Altos, California: Morgan Kaufmann Publishers, Inc.

Moreno, A. 1998. Avoiding logical omniscience and perfect reasoning: a survey. *AI Communications* 11:101–122.

Moses, Y. 1988. Resource-bounded knowledge. In Vardi, M. Y., ed., *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the Second Conference on Theoretical Aspects of Reasoning About Knowledge*, 261–275. San Francisco: Morgan Kaufmann.

Nirkhe, M.; Kraus, S.; and Perlis, D. 1994. Thinking takes time: a modal active-logic for reasoning in time. Technical Report CS-TR-3249, University of Maryland, College Park.

Parikh, R., ed. 1990. *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the Third Conference (TARK 1990)*. San Mateo, CA, USA: Morgan Kaufmann.

Pease, M.; Shostak, R.; and Lamport, L. 1980. Reaching agreement in the presence of faults. *J. ACM* 27(2):228–234.

Scott, D. S. 1970. Advice on modal logic. In Lambert, K., ed., *Philosophical Problems in Logic*. Dordrecht: D. Reidel Publishing Co. 143–173.

Sim, K. M. 1997. Epistemic logic and logical omniscience: A survey. *International Journal of Intelligent Systems* 12:57–81.

van der Hoek, W., and Wooldridge, M. 2002. Tractable multiagent planning for epistemic goals. In *Proceedings of the First International Conference on Autonomous Agents and Multiagent Systems (AAAMAS-02)*.

van der Hoek, W., and Wooldridge, M. 2003. Cooperation, knowledge and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica* 75:125–157.

Vardi, M. Y. 1986. On epistemic logic and logical omniscience. In Halpern, J. Y., ed., *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the First Conference*, 293–305. Los Altos, California: Morgan Kaufmann Publishers, Inc.

Wooldridge, M. 1995a. An abstract general model and logic of resource-bounded believers. In Cox, M., and Freed, M., eds., *Representing Mental States and Mechanisms — Proceedings of the 1995 AAAI Spring Symposium*, 136–141. AAAI Press.

Wooldridge, M. 1995b. Temporal belief logics for modelling distributed artificial intelligence systems. In O'Hare, G. M. P., and Jennings, N. R., eds., *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons. chapter 10.

Part V

Appendices

Appendix A

A Fixed-point Characterization of *EL*

The definitions of AL (Def. 3.5), $TL(AL)$ (Def. 3.2) and EL (Def. 3.6) are collected in Definition A.1 below.

In the following I write just TL for $TL(AL)$. Also, I shall be pedantic with respect to the notation for basic terms and use the underlined notation.

The three languages are defined over the alphabet

$$A = \Theta \cup \{\triangle_i, \triangledown_i, \neg, \wedge, (,), \doteq\, \underline{\{}, \underline{\}}, \underline{,}\} \tag{A.1}$$

I use the following notation for the set of descriptors for finite subsets of a set $X \subseteq A^*$:

$$Fin(X) = \{\underline{\{}\alpha_1\underline{,} \ldots \underline{,}\alpha_k\underline{\}} : \alpha_i \in X, k \geq 1\} \tag{A.2}$$

Definition A.1

- TL is the least set such that
  - $Fin(AL) \subseteq TL$
  - If $T, U \in TL$ then $\left.\begin{array}{c}(T \sqcup U)\\(T \sqcap U)\end{array}\right\} \in TL$
- AL is the least set such that
  - $\Theta \subseteq AL$
  - If $T \in TL$ then $\left.\begin{array}{c}\triangle_i T\\\triangledown_i T\end{array}\right\} \in AL$ $(1 \leq i \leq n)$
  - If $\alpha, \beta \in AL$ then $\left.\begin{array}{c}\neg\alpha\\(\alpha \wedge \beta)\end{array}\right\} \in AL$
- EL is the least set such that
  - $AL \subseteq EL$
  - If $T, U \in TL$ then $(T \doteq U) \in EL$
  - If $\phi, \psi \in EL$ then $\left.\begin{array}{c}\neg\phi\\(\phi \wedge \psi)\end{array}\right\} \in EL$ $\square$

I show that the three sets are well-defined.

Lemma A.1 The recursive Definition A.1 has a solution. □

PROOF I show that the associated generating function has a fixed point.

To make the discussion easier, Definition A.1 is rewritten as a set of three mutually recursive equations:

$$TL = Fin(AL) \cup \{T \sqcup U : T, U \in TL\} \cup \{T \sqcap U : T, U \in TL\} \tag{A.3}$$

$$\begin{aligned} AL =& \Theta \cup \{\triangle_i T : T \in TL\} \cup \{\triangledown_i T : T \in TL\} \cup \\ & \{\neg\phi : \phi \in AL\} \cup \{(\phi \wedge \psi) : \phi, \psi \in AL\} \end{aligned} \tag{A.4}$$

$$\begin{aligned} EL =& AL \cup \{T \doteq U : T, U \in TL\} \cup \\ & \{\neg\phi : \phi \in EL\} \cup \{(\phi \wedge \psi) : \phi, \psi \in EL\} \end{aligned} \tag{A.5}$$

The function

$$g : \wp(A^*)^3 \to \wp(A^*)^3 \tag{A.6}$$

is defined as a generating function corresponding to the recursive definitions, as follows (the names of the variables have been changed):

$$g(TL', AL', EL') = \langle g_T(TL', AL', EL'), g_O(TL', AL', EL'), g_E(TL', AL', EL') \rangle \tag{A.7}$$

where

$$g_T(TL', AL', EL') = Fin(AL') \cup \{T \sqcup U : T, U \in TL'\} \cup \{T \sqcap U : T, U \in TL'\} \tag{A.8}$$

$$\begin{aligned} g_O(TL', AL', EL') =& \Theta \cup \{\triangle_i T : T \in TL'\} \cup \{\triangledown_i T : T \in TL'\} \cup \\ & \{\neg\phi : \phi \in AL'\} \cup \{(\phi \wedge \psi) : \phi, \psi \in AL'\} \end{aligned} \tag{A.9}$$

$$\begin{aligned} g_E(TL', AL', EL') =& AL' \cup \{T \doteq U : T, U \in TL'\} \cup \\ & \{\neg\phi : \phi \in EL'\} \cup \{(\phi \wedge \psi) : \phi, \psi \in EL'\} \end{aligned} \tag{A.10}$$

Then, an ordering relation $\sqsubseteq$ on $\wp(A^*)^3$ is defined:

$$\langle T, O, E \rangle \sqsubseteq \langle T', O', E' \rangle \Leftrightarrow \left\{ \begin{array}{ll} T \subseteq T' & \text{and} \\ O \subseteq O' & \text{and} \\ E \subseteq E' & \end{array} \right. \tag{A.11}$$

$\sqsubseteq$ is a pointed (has bottom) complete partial order, so if we can prove that g is continuous with respect to $\sqsubseteq$ we have proved that it has a least fixed point.

Define the least upper bound $c_1 \sqcup_{\sqsubset} c_2$ of two elements $c_1, c_2 \in \wp(A^*)^3$ in the usual way. It is easy to see that, for $S \subseteq \wp(A^*)^3$

$$\bigsqcup_{\sqsubset} S = \langle \bigcup\{T : \langle T, O, E \rangle \in S\}, \bigcup\{O : \langle T, O, E \rangle \in S\}, \bigcup\{E : \langle T, O, E \rangle \in S\} \rangle \tag{A.12}$$

g is continuous iff for every chain C in $\wp(A^*)^3$,

$$g(\bigsqcup_{\sqsubset} C) = \bigsqcup_{\sqsubset}\{g(c) : c \in C\} \tag{A.13}$$

Henceforth, let $C = \{c_1, c_2, \ldots\}$ where $c_i \sqsubseteq c_{i+1}$ be an arbitrary chain in $\wp(A^*)^3$[1], and let $c_i = \langle T_i, O_i, E_i \rangle$. Clearly, $\{T_i : c_i \in C\}$, $\{O_i : c_i \in C\}$ and $\{E_i : c_i \in C\}$ are all chains in $\wp(A^*)$, and $T_i \subseteq T_{i+1}$, $O_i \subseteq O_{i+1}$ and $E_i \subseteq E_{i+1}$.

Let $S = \{S_1, S_2, \ldots\}$ be a chain in $\wp(A^*)$. I show two properties of S. The first is:

$$Fin(\bigcup_i \{S_i\}) = \bigcup_i \{Fin(S_i)\} \tag{A.14}$$

I show that this equation holds. First, let $x \in \bigcup_i \{Fin(S_i)\}$. Then, $x \in Fin(S_j)$ for some j, and $x = \{\alpha_1, \ldots, \alpha_k\}$ where $\alpha_i \in S_j$. Since $S_j \subseteq \bigcup_i \{S_i\}$, $\alpha_i \in \bigcup_i \{S_i\}$ and thus $x \in Fin(\bigcup_i \{S_i\})$. Second, let $x = \{\alpha_1, \ldots, \alpha_k\} \in Fin(\bigcup_i \{S_i\})$. Then $\alpha_i \in \bigcup_i \{S_i\}$, so for each α_i there is a S'_i such that $\alpha \in S'_i$.

The chain $S'_1, \ldots, S'_k$ is finite, so it contains it's least upper bound – let the lub be S'_j. Since $S'_1 \cup \ldots \cup S'_k = S'_j$, we have that $\alpha_i \subseteq S'_j$ for all i and so $x \in Fin(S'_j)$. Since $S'_j \in S$, $x \in \bigcup_i \{Fin(S_i)\}$.

The second property of the chain S we need is:

$$x_1, x_2 \in \bigcup_i \{S_i\} \text{ iff there is some } S_j \text{ such that } x_1, x_2 \in S_j \tag{A.15}$$

If $x_1, x_2 \in \bigcup_i \{S_i\}$ then there exist S_1, S_2 such that $x_1 \in S_1$ and $x_2 \in S_2$. If $S_1 \subseteq S_2$ then $x_1, x_2 \in S_2$ and if $S_2 \subseteq S_1$ then $x_1, x_2 \in S_1$. Conversely, if $x_1, x_2 \in S_j$ for some S_j then $x_1, x_2 \in \bigcup_i \{S_i\}$ trivially.

In order to show eq. A.13, I first show that

$$g_T(\bigsqcup_{\sqsubseteq} C) = \bigcup \{g_T(c) : c \in C\} \tag{A.16}$$

[1]A chain in $\wp(A^*)^3$ is countable.

$$\begin{aligned}
g_T(\bigsqcup_{\sqsubset} C) &= g_T(\bigcup_i \{T_i\}, \bigcup_i \{O_i\}, \bigcup_i \{E_i\}) && \text{By (A.12)} \quad \text{(A.17)}\\
&= Fin(\bigcup_i \{O_i\}) \cup \{T \sqcup U : T, U \in \bigcup_i \{T_i\}\}\\
&\quad \cup \{T \sqcap U : T, U \in \bigcup_i \{T_i\}\} && \text{(A.18)}\\
&= \bigcup_i \{Fin(O_i)\} \cup \{T \sqcup U : T, U \in \bigcup_i \{T_i\}\}\\
&\quad \cup \{T \sqcap U : T, U \in \bigcup_i \{T_i\}\} && \text{By (A.14)} \quad \text{(A.19)}\\
&= \bigcup_i \{Fin(O_i)\} \cup \{T \sqcup U : T, U \in T_j \text{for some } T_j\}\\
&\quad \cup \{T \sqcap U : T, U \in T_j \text{for some } T_j\} && \text{By (A.15)} \quad \text{(A.20)}\\
&= \bigcup_i \{Fin(O_i)\} \cup \bigcup_i \{T \sqcup U : T, U \in T_i\} && \text{(A.21)}\\
&\quad \cup \bigcup_i \{T \sqcap U : T, U \in T_i\} && \text{(A.22)}\\
&= \bigcup \{Fin(O_i) \cup \{T \sqcup U : T, U \in T_i\}\\
&\quad \cup \{T \sqcap U : T, U \in T_i\} : \langle T_i, O_i, E_i \rangle \in C\} && \text{(A.23)}\\
&= \bigcup \{g_T(c) : c \in C\} && \text{(A.24)}
\end{aligned}$$

Second, I show that

$$g_O(\bigsqcup_{\sqsubset} C) = \bigcup \{g_O(c) : c \in C\} \tag{A.25}$$

$$\begin{aligned}
g_O(\bigsqcup_{\sqsubset} C) &= g_O(\bigcup_i \{T_i\}, \bigcup_i \{O_i\}, \bigcup_i \{E_i\}) && \text{By (A.12)}\\
&= \Theta \cup \{\triangle_i T : T \in \bigcup_i \{T_i\}\} \cup \{\triangledown_i T : T \in \bigcup_i \{T_i\}\} \cup\\
&\quad \{\neg\phi : \phi \in \bigcup_i \{O_i\}\} \cup \{(\phi \wedge \psi) : \phi, \psi \in \bigcup_i \{O_i\}\} && \text{(A.26)}\\
&= \Theta \cup \{\triangle_i T : T \in \bigcup_i \{T_i\}\} \cup \{\triangledown_i T : T \in \bigcup_i \{T_i\}\} \cup\\
&\quad \{\neg\phi : \phi \in \bigcup_i \{O_i\}\} \cup \{(\phi \wedge \psi) : \phi, \psi \in O_j \text{for some } O_j\} && \text{By (A.15)}\\
&= \Theta \cup \bigcup_i \{\triangle_i T : T \in T_i\} \cup \bigcup_i \{\triangledown_i T : T \in T_i\} \cup\\
&\quad \bigcup_i \{\neg\phi : \phi \in O_i\} \cup \bigcup_i \{(\phi \wedge \psi) : \phi, \psi \in O_i\} && \text{(A.27)}\\
&= \bigcup \{\Theta \cup \{\triangle_i T : T \in T_i\} \cup \{\triangledown_i T : T \in T_i\} \cup\\
&\quad \{\neg\phi : \phi \in O_i\} \cup \{(\phi \wedge \psi) : \phi, \psi \in O_i\} : \langle T_i, O_i, E_i \rangle \in C\\
& && \text{(A.28)}\\
&= \bigcup \{g_O(c) : c \in C\} && \text{(A.29)}
\end{aligned}$$

Third, I show that

$$g_E(\bigsqcup_{\sqsubset} C) = \bigcup\{g_E(c) : c \in C\} \tag{A.30}$$

$$g_E(\bigsqcup_{\sqsubset} C) = g_E(\bigcup_i\{T_i\}, \bigcup_i\{O_i\}, \bigcup_i\{E_i\}) \qquad \text{By (A.12)}$$

$$\begin{aligned} &= \bigcup_i\{O_i\} \cup \{T \doteq U : T, U \in \bigcup_i\{T_i\}\} \cup \\ &\quad \{\neg\phi : \phi \in \bigcup_i\{E_i\}\} \cup \{(\phi \wedge \psi) : \phi, \psi \in \bigcup_i\{E_i\}\} \end{aligned} \tag{A.31}$$

$$\begin{aligned} &= \bigcup_i\{O_i\} \cup \{T \doteq U : T, U \in T_j \text{for some } T_j\} \cup \\ &\quad \{\neg\phi : \phi \in \bigcup_i\{E_i\}\} \cup \{(\phi \wedge \psi) : \phi, \psi \in E_j \text{for some } E_j\} \end{aligned} \tag{A.32}$$

$$\begin{aligned} &= \bigcup_i\{O_i\} \cup \bigcup_i\{T \doteq U : T, U \in T_i\} \cup \\ &\quad \bigcup_i\{\neg\phi : \phi \in E_i\} \cup \bigcup_i\{(\phi \wedge \psi) : \phi, \psi \in E_i\} \end{aligned} \tag{A.33}$$

$$\begin{aligned} &= \bigcup\{O_i \cup \{T \doteq U : T, U \in T_i\} \cup \\ &\qquad \{\neg\phi : \phi \in E_i\} \cup \{(\phi \wedge \psi) : \phi, \psi \in E_i\} : \langle T_i, O_i, E_i\rangle \in C\} \end{aligned} \tag{A.34}$$

$$= \bigcup\{g_E(c) : c \in C\} \tag{A.35}$$

Finally, I show eq. A.13:

$$g(\bigsqcup_{\sqsubset} C) = \langle g_T(\bigsqcup_{\sqsubset} C), g_O(\bigsqcup_{\sqsubset} C), g_E(\bigsqcup_{\sqsubset} C)\rangle \tag{A.36}$$

$$= \langle \bigcup\{g_T(c) : c \in C\}, \bigcup\{g_O(c) : c \in C\}, \bigcup\{g_E(c) : c \in C\}\rangle \tag{A.37}$$

$$\text{By (A.16),(A.25),(A.30)} \tag{A.38}$$

$$= \bigsqcup_{\sqsubset}\{g(c) : c \in C\} \qquad \text{By (A.12)}$$

Thus g has a (least) fixed point, which is a solution to the recursive definition. ■

Corollary A.1 (Fixed Point Characterization of *TL*, *AL* and *EL*) Let g be as defined in the proof of Lemma A.1.

$$\langle TL, AL, EL\rangle = \bigsqcup_{\sqsubset}\{g^n(\emptyset, \emptyset, \emptyset) : n \geq 0\} \tag{A.39}$$

□

Appendix B

Proof of Lemma 4.1

To save space, I only show proofs for selected parts of the lemma. Formally, Corollary 4.2 shows that the term calculus is complete and hence that the rest of the lemma also holds. All parts of the lemma which is used in the proof of Corollary 4.2 are proven below.

1.

1	$\vdash T \sqcap (T \sqcup U) \doteq T$	T11
2	$\vdash T \doteq T \sqcap (T \sqcup U)$	**MP**(T2,1)
3	$\vdash T \doteq T$	T1
4	$\vdash T \sqcup T \doteq T \sqcup (T \sqcap (T \sqcup U))$	**MP**(T4,2,3)
5	$\vdash T \sqcup (T \sqcap (T \sqcup U)) \doteq T$	T10
6	$\vdash T \sqcup T \doteq T \sqcup (T \sqcap (T \sqcup U)) \wedge T \sqcup (T \sqcap (T \sqcup U)) \doteq T$	**MP**(**Prop**,4,5)
7	$\vdash T \sqcup T \doteq T$	**MP**(T3,6)

2.

1	$\vdash T \sqcup (T \sqcap U) \doteq T$	T10
2	$\vdash T \doteq T \sqcup (T \sqcap U)$	**MP**(T2,1)
3	$\vdash T \doteq T$	T1
4	$\vdash T \sqcap T \doteq T \sqcap (T \sqcup (T \sqcap U))$	**MP**(T5,2,3)
5	$\vdash T \sqcap (T \sqcup (T \sqcap U)) \doteq T$	T11
6	$\vdash T \sqcap T \doteq T \sqcap (T \sqcup (T \sqcap U)) \wedge T \sqcap (T \sqcup (T \sqcap U)) \doteq T$	**MP**(**Prop**,4,5)
7	$\vdash T \sqcap T \doteq T$	**MP**(T3,6)

12.

1	$\vdash T \doteq U \sqcup T \wedge U \sqcup T \doteq U \to T \doteq U$	T3
2	$\vdash U \sqcup T \doteq T \to T \doteq U \sqcup T$	T2
3	$\vdash T \sqcup U \doteq U \to U \sqcup T \doteq U$	**MP**(**Prop**,T4,T6)
4	$\vdash T \sqcup U \doteq U \wedge U \sqcup T \doteq T \to U \sqcup T \doteq U \wedge T \doteq U \sqcup T$	**MP**(**Prop**,2,3)
5	$\vdash T \sqcup U \doteq U \wedge U \sqcup T \doteq T \to T \doteq U$	**MP**(T3,1,4)
6	$\vdash T \doteq U \to T \sqcup U \doteq U \sqcup U$	**MP**(**Prop**,T4,T1)
7	$\vdash U \sqcup U \doteq U$	L.4.1.1
8	$\vdash T \doteq U \to T \sqcup U \doteq U$	**MP**(**Prop**,T3,6,7)
9	$\vdash U \doteq T \to U \sqcup T \doteq T \sqcup T$	**MP**(**Prop**,T4,T1)
10	$\vdash T \sqcup T \doteq T$	L.4.1.1
11	$\vdash U \doteq T \to U \sqcup T \doteq T$	**MP**(**Prop**,T3,9,10)
12	$\vdash T \doteq U \to U \doteq T$	T2
13	$\vdash T \doteq U \to U \sqcup T \doteq T$	**MP**(T3,11,12)
14	$\vdash T \doteq U \leftrightarrow T \sqcup U \doteq U \wedge U \sqcup T \doteq T$	**MP**(**Prop**,5,8,13)

13.

1	$\vdash (T \sqcup U) \sqcap (T \sqcup V) \doteq ((T \sqcup U) \sqcap T) \sqcup ((T \sqcup U) \sqcap V)$	T12
2	$\vdash T \sqcap (T \sqcup U) \doteq T$	T11
3	$\vdash (T \sqcup U) \sqcap V \doteq V \sqcap (T \sqcup U)$	T7
4	$\vdash V \sqcap (T \sqcup U) \doteq (V \sqcap T) \sqcup (V \sqcap U)$	T12
5	$\vdash (T \sqcup U) \sqcap V \doteq (V \sqcap T) \sqcup (V \sqcap U)$	**MP**(T3,3,4)
6	$\vdash ((T \sqcup U) \sqcap T) \sqcup ((T \sqcup U) \sqcap V) \doteq T \sqcup ((V \sqcap T) \sqcup (V \sqcap U))$	**MP**(T4,2,5)
7	$\vdash (T \sqcup U) \sqcap (T \sqcup V) \doteq T \sqcup ((V \sqcap T) \sqcup (V \sqcap U))$	**MP**(T3,1,6)
8	$\vdash T \sqcup ((V \sqcap T) \sqcup (V \sqcap U)) \doteq (T \sqcup (V \sqcap T)) \sqcup (V \sqcap U)$	T8
9	$\vdash (T \sqcup U) \sqcap (T \sqcup V) \doteq (T \sqcup (V \sqcap T)) \sqcup (V \sqcap U)$	**MP**(T3,7,8)
10	$\vdash V \sqcap T \doteq T \sqcap V$	T2
11	$\vdash T \doteq T$	T1
12	$\vdash T \sqcup (V \sqcap T) \doteq T \sqcup (T \sqcap V)$	**MP**(T4,10,11)
13	$\vdash T \sqcup (T \sqcap V) \doteq T$	T10
14	$\vdash T \sqcup (V \sqcap T) \doteq T$	**MP**(T3,12,13)
15	$\vdash V \sqcap U \doteq U \sqcap V$	T2
16	$\vdash (T \sqcup (V \sqcap T)) \sqcup (V \sqcap U) \doteq T \sqcup (U \sqcap V)$	**MP**(T4,14,15)
17	$\vdash (T \sqcup U) \sqcap (T \sqcup V) \doteq T \sqcup (U \sqcap V)$	**MP**(T3,9,16)
18	$\vdash T \sqcup (U \sqcap V) \doteq (T \sqcup U) \sqcap (T \sqcup V)$	**MP**(T2,17)

Appendix C

Proof of Lemma 4.5

I prove both parts of the lemma simultaneously, by induction over the degree of $T_1 \sqcup T_2$ (of course, $d(T_1) \leq d(T_1 \sqcup T_2) \geq d(T_2)$).

Induction basis (Lemma 4.5) For the induction basis, let $d(T_1 \sqcup T_2) = 1$ – in which case $d(T_1) = d(T_2) = 1$. I first show an intermediate result:

$$\text{If } d(\{\alpha\}) = d(\{\beta\}) = 1 \text{ and } [\alpha] \neq [\beta] \text{ then } \vdash \neg(\{\alpha\} \doteq \{\beta\}) \tag{C.1}$$

Let $d(\{\alpha\}) = d(\{\beta\}) = 1$ and $[\alpha] \neq [\beta]$. I show (C.1) by structural induction over α. Note that since $\{\alpha\}$ and $\{\beta\}$ are of degree 1, neither α or β can contain subformulas $\triangle_i T$ or $\triangledown_i T$.

Induction basis (C.1) For the single base case, let $\alpha = p$. Then $[\alpha] = p$. Assume that $[\beta] \neq p$. Then $\beta \neq p$, and

$$\vdash \neg(\{\alpha\} \doteq \{\beta\}) \tag{C.2}$$

holds by N2.

Induction step (C.1) For the induction step, first let $\alpha = \neg\gamma$ – then $d(\{\gamma\}) = 1$ and $[\alpha] = \neg[\gamma]$. The induction hypothesis is that, for all γ' with $d(\{\gamma'\}) = 1$, if $[\gamma] \neq [\gamma']$ then $\vdash \neg(\{\gamma\} \doteq \{\gamma'\})$. $[\beta] \neq \neg[\gamma]$. If β starts with a negation, $\beta = \neg\beta'$ where $d(\{\beta'\}) = 1$ and $[\gamma] \neq [\beta']$. Then $\vdash \neg(\{\gamma\} \doteq \{\beta'\})$ by the induction hypothesis, and (C.2) follows by N7. If β does not start with a negation, (C.2) follows by N8. For the second part of the induction step, let $\alpha = (\gamma_1 \wedge \gamma_2)$. Then $[\alpha] = ([\gamma_1] \wedge [\gamma_2])$. $[\beta] \neq ([\gamma_1] \wedge [\gamma_2])$. If β is not a conjunction, then (C.2) follows by N10. If $\beta = (\beta_1 \wedge \beta_2)$ then $([\gamma_1] \wedge [\gamma_2]) \neq ([\beta_1] \wedge [\beta_2])$. Then either $[\gamma_1] \neq [\beta_1]$ or $[\gamma_2] \neq [\beta_2]$ and either $\vdash \neg(\{\gamma_1\} \doteq \{\beta_1\})$ or $\vdash \neg(\{\gamma_2\} \doteq \{\beta_2\})$ holds by the induction hypothesis, and then (C.2) follows by N9.

I now show that, if $\{\beta_j\}$ ($1 \leq j \leq l$) are terms,

$$\text{If } d(T) = 1 \text{ and } [T] = \{[\beta_1], \ldots, [\beta_l]\} \text{ then } \vdash T \doteq \{\beta_1, \ldots, \beta_l\} \tag{C.3}$$

by structural induction over T. I make use of the fact that for any term S of degree 1, if $[S] = \{[\alpha_1], \ldots, [\alpha_k]\}$ then $[\alpha_j] = \alpha_j$ ($1 \leq j \leq k$) (because

$d(\{\alpha_j\}) = 1$; α_j cannot contain any knowledge operators) and thus $[S] = \{\alpha_1, \dots, \alpha_k\}$[1]. Let [T] = $\{[\beta_1], \dots, [\beta_l]\} = \{\beta_1, \dots, \beta_l\}$.

Induction basis (C.3) For the induction basis, let $T = \{\alpha_1, \dots, \alpha_k\}$. Then $\{[\beta_1], \dots, [\beta_l]\} = [T] = \{[\alpha_1], \dots, [\alpha_k]\}$, $\{\beta_1, \dots, \beta_l\} = \{\alpha_1, \dots, \alpha_k\}$ and $\vdash \{\alpha_1, \dots, \alpha_k\} \doteq \{\beta_1, \dots, \beta_l\}$ by Lemma 4.2.

Induction step (C.3) For the induction step, first consider the case when $T = U \sqcup V$. $[U] \cup [V] = \{[\beta_1], \dots, [\beta_l]\}$, so $[U] = \{[\beta_1^U], \dots, [\beta_{k_U}^U]\}$ and $[V] = \{[\beta_1^V], \dots, [\beta_{k_V}^V]\}$, where each $\beta_j^U, \beta_j^V \in \{\beta_1, \dots, \beta_l\}$.

1	$\vdash U \doteq \{\beta_1^U, \dots, \beta_{k_U}^U\}$	Ind. hyp.
2	$\vdash V \doteq \{\beta_1^V, \dots, \beta_{k_V}^V\}$	Ind. hyp.
3	$\vdash U \sqcup V \doteq \{\beta_1^U, \dots, \beta_{k_U}^U\} \sqcup V$	**Repl**,1
4	$\vdash \{\beta_1^U, \dots, \beta_{k_U}^U\} \sqcup V \doteq \{\beta_1^U, \dots, \beta_{k_U}^U\} \sqcup \{\beta_1^V, \dots, \beta_{k_V}^V\}$	**Repl**,2
5	$\vdash U \sqcup V \doteq \{\beta_1^U, \dots, \beta_{k_U}^U\} \sqcup \{\beta_1^V, \dots, \beta_{k_V}^V\}$	**MP**,T3,3,4

From several applications of T13, and **Repl**, we have that

6	$\vdash \{\beta_1^U, \dots, \beta_{k_U}^U\} \sqcup \{\beta_1^V, \dots, \beta_{k_V}^V\} \doteq \{\beta_1^U, \dots, \beta_{k_U}^U, \beta_1^V, \dots, \beta_{k_V}^V\}$	
7	$\vdash U \sqcup V \doteq \{\beta_1^U, \dots, \beta_{k_U}^U, \beta_1^V, \dots, \beta_{k_V}^V\}$	**MP**,T3,5,6

The sequence $\beta_1^U, \dots, \beta_{k_U}^U, \beta_1^V, \dots, \beta_{k_V}^V$ is a permutation, possibly with duplicates, of the sequence $\beta_1, \dots, \beta_l$, so

$$\vdash U \sqcup V \doteq \{\beta_1, \dots, \beta_l\}$$

by Lemma 4.2 and T3. For the second case in the induction step, let $T = U \sqcap V$ where $[U] = \{[\alpha_1^U], \dots, [\alpha_{k_U}^U]\} = \{\alpha_1^U, \dots, \alpha_{k_U}^U\}$ and $[V] = \{[\alpha_1^V], \dots, [\alpha_{k_V}^V]\} = \{\alpha_1^V, \dots, \alpha_{k_V}^V\}$. In order to prove this case, I first show that

$$\vdash \{\alpha_1^U, \dots, \alpha_{k_U}^U\} \sqcap \{\alpha_1^V, \dots, \alpha_{k_V}^V\} \doteq \{\beta_1, \dots, \beta_l\} \tag{C.4}$$

We have that

$$\begin{aligned} &\vdash \{\alpha_1^U, \dots, \alpha_{k_U}^U\} \sqcap \{\alpha_1^V, \dots, \alpha_{k_V}^V\} \doteq \\ &\quad (\{\alpha_1^U\} \sqcap \{\alpha_1^V\}) \sqcup (\{\alpha_1^U\} \sqcap \{\alpha_2^V\}) \sqcup \cdots \sqcup (\{\alpha_1^U\} \sqcap \{\alpha_{k_V}^V\}) \sqcup \\ &\quad (\{\alpha_2^U\} \sqcap \{\alpha_1^V\}) \sqcup (\{\alpha_2^U\} \sqcap \{\alpha_2^V\}) \sqcup \cdots \sqcup (\{\alpha_2^U\} \sqcap \{\alpha_{k_V}^V\}) \sqcup \\ &\quad \vdots \\ &(\{\alpha_{k_U}^U\} \sqcap \{\alpha_1^V\}) \sqcup (\{\alpha_{k_U}^U\} \sqcap \{\alpha_2^V\}) \sqcup \cdots \sqcup (\{\alpha_{k_U}^U\} \sqcap \{\alpha_{k_V}^V\}) \end{aligned} \tag{C.5}$$

[1]Recall that S is a *term* and an expression like $S = \{\alpha_1, \dots, \alpha_k\}$ is a shorthand for $S = \underline{\{}\alpha_1\underline{,} \dots \underline{,}\alpha_k\underline{\}}$, while $[S]$ is a *set* and an expression like $[S] = \{\alpha_1, \dots, \alpha_k\}$ denotes the set consisting of the elements $\alpha_1, \dots, \alpha_k$. It is only a formula which does not contain a knowledge operator, e.g. all formulae in a term of degree 1, that has an interpretation equal to the formula itself.

by T13 and repeated applications of T12, **Repl** and the axioms of equality. The above expression contains exactly one term on the form $\{\alpha^U\} \sqcap \{\alpha^V\}$ for each pair $\alpha^U \in [U], \alpha^V \in [V]$. If $\beta \in [T]$, then $\alpha_i^U = \beta \in [U]$ and $\alpha_j^V = \beta \in [V]$ and we have $\vdash \{\alpha_i^U\} \doteq \{\alpha_j^V\}$ by T1 and thus $\vdash \{\alpha_i^U\} \sqcap \{\alpha_j^V\} \doteq \{\beta\}$ by T14. If $\alpha_i^U \in [U]$ but $\alpha_i^U \notin [T]$, for every $\alpha_j^V \in [V]$ $\alpha_i^U \neq \alpha_j^V$, and because $[\alpha_i^U] = \alpha_i^U$ and $[\alpha_j^V] = \alpha_j^V$ ($\{\alpha_i^U\}, \{\alpha_j^V\}$ are of degree 1), $\vdash \neg(\{\alpha_i^U\} \doteq \{\alpha_j^V\})$ by (C.1) and thus $\vdash \{\alpha_i^U\} \sqcap \{\alpha_j^V\} \doteq \emptyset$ by T15. Similarly, if $\alpha_j^V \in [V]$ but $\alpha_j^V \notin [T]$, $\vdash \{\alpha_i^U\} \sqcap \{\alpha_j^V\} \doteq \emptyset$ for every $\alpha_i^U \in [U]$. Then, (C.4) follows by (C.5), **Repl** and T13. I now use (C.4) to show the current induction step:

1	$\vdash U \doteq \{\alpha_1^U, \ldots, \alpha_{k_U}^U\}$	Ind. hyp.
2	$\vdash V \doteq \{\alpha_1^V, \ldots, \alpha_{k_V}^V\}$	Ind. hyp.
3	$\vdash U \sqcap V \doteq \{\alpha_1^U, \ldots, \alpha_{k_U}^U\} \sqcap \{\alpha_1^V, \ldots, \alpha_{k_V}^V\}$	**MP**,T5,1,2
4	$\vdash \{\alpha_1^U, \ldots, \alpha_{k_U}^U\} \sqcap \{\alpha_1^V, \ldots, \alpha_{k_V}^V\} \doteq \{\beta_1, \ldots, \beta_l\}$	(C.4)
5	$\vdash U \sqcap V \doteq \{\beta_1, \ldots, \beta_l\}$	**MP**,T3,3,4

This completes the proof of (C.3).

I can now show the base case in the inductive proof over the degree of $T_1 \sqcup T_2$. Let α_j, β_j be such that $[T_1] = \{[\alpha_1], \ldots, [\alpha_k]\}$ and $[T_2] = \{[\beta_1], \ldots, [\beta_m]\}$. First, consider the case that $[T_1] = [T_2]$:

1	$\vdash T_1 \doteq \{\alpha_1, \ldots, \alpha_k\}$	(C.3)
2	$\vdash T_2 \doteq \{\alpha_1, \ldots, \alpha_k\}$	(C.3)
3	$\vdash \{\alpha_1, \ldots, \alpha_k\} \doteq T_2$	**MP**,T2,2
4	$\vdash T_1 \doteq T_2$	**MP**,T3,1,3

Second, consider the case that $[T_1] \neq [T_2]$. I show that

$$\vdash \neg(\{\alpha_1, \ldots, \alpha_k\} \doteq \{\beta_1, \ldots, \beta_m\}) \tag{C.6}$$

I assume that there is an $[\alpha_i] \in [T_1]$ such that $[\alpha_i] \notin [T_2]$ (the proof is equivalent in the case that $[\beta_i] \in [T_2]$ and $[\beta_i] \notin [T_1]$). Since $[\alpha_i] \neq [\beta_j]$ ($1 \leq j \leq m$), $\vdash \neg(\{\alpha_i\} \doteq \{\beta_j\})$ by (C.1) and $\vdash \neg((\{\alpha_i\} \doteq \{\beta_1\}) \vee \cdots \vee (\{\alpha_i\} \doteq \{\beta_m\}))$ by **Prop**. By **Prop** and N1, $\vdash \neg(\{\alpha_1, \ldots, \alpha_k\} \preceq \{\beta_1, \ldots, \beta_m\})$ and (C.6) follows by Lemma 4.1.12. By (C.3) $\vdash \{\alpha_1, \ldots, \alpha_k\} \doteq T_1$, and by (C.6), T3 and **Prop**

$$\vdash \neg(T_1 \doteq \{\beta_1, \ldots, \beta_m\}) \tag{C.7}$$

By (C.3) $\vdash \{\beta_1, \ldots, \beta_m\} \doteq T_2$, and by (C.7), T3, **Prop** and T2,

$$\vdash \neg T_1 \doteq T_2$$

This completes the base case in the inductive proof of the lemma.

Induction step (Lemma 4.5) For the induction step, let the lemma hold for all terms T_1, T_2 such that $d(T_1 \sqcup T_2) \leq k$. Let $d(T_1 \sqcup T_2) = k + 1$. I now show two intermediate results. I first show that

$$\text{If } d(\{\alpha\}) \leq k+1 \geq d(\{\beta\}) \text{ and } [\alpha] = [\beta] \text{ then } \vdash \{\alpha\} \doteq \{\beta\} \tag{C.8}$$

for all β by structural induction over α. Let $d(\{\alpha\}) \leq k+1 \geq d(\{\beta\})$ and $[\alpha] = [\beta]$.

Induction basis (C.8) For the first base case, let $\alpha = p \in \Theta$. Then $\beta = p$, and

$$\vdash \{\alpha\} \doteq \{\beta\} \tag{C.9}$$

follows by T1. For the second base case, let $\alpha = \triangle_i S_1$. Then $\beta = \triangle_i S_2$ where $[S_1] = [S_2]$. $d(S_1) \leq k \geq d(S_2)$, so $d(S_1 \sqcup S_2) \leq k$ and $\vdash S_1 \doteq S_2$ holds by the induction hypothesis (in the "outmost" inductive proof). (C.9) follows by N3. Similarly, in the third base case when $\alpha = \triangledown_i S_1$ and $\beta = \triangledown_i S_2$, (C.9) follows by the induction hypothesis and N5.

Induction step (C.8) For the induction step, first let $\alpha = \neg\gamma$ – then $[\alpha] = \neg[\gamma]$. The induction hypothesis is that, for all γ', if $[\gamma] = [\gamma']$ then $\vdash \{\gamma\} \doteq \{\gamma'\}$. β must start with a negation; say $\beta = \neg\beta'$ where $[\beta'] = [\gamma]$. $\vdash \{\gamma\} \doteq \{\beta'\}$ by the induction hypothesis, and (C.9) follows by N7. For the second part of the induction step, let $\alpha = (\gamma_1 \wedge \gamma_2)$. Then $[\alpha] = ([\gamma_1] \wedge [\gamma_2])$. β must be a conjunction; say $\beta = (\beta_1 \wedge \beta_2)$ where $[\beta_1] = [\gamma_1]$ and $[\beta_2] = [\gamma_2]$. $\vdash \{\gamma_1\} \doteq \{\beta_1\}$ and $\vdash \{\gamma_2\} \doteq \{\beta_2\}$ by the induction hypothesis, and (C.9) follows by N9.

Second, I show a general version of (C.1):

$$\text{If } d(\{\alpha\}) \leq k+1 \geq d(\{\beta\}) \text{ and } [\alpha] \neq [\beta] \text{ then } \vdash \neg\{\alpha\} \doteq \{\beta\} \tag{C.10}$$

for all β by structural induction over α. Let $d(\{\alpha\}) \leq k+1 \geq d(\{\beta\})$ and $[\alpha] \neq [\beta]$.

Induction basis (C.10) For the first base case, let $\alpha = p$. Then $[\alpha] = p$. Then $\beta \neq p$, and

$$\vdash \neg(\{\alpha\} \doteq \{\beta\}) \tag{C.11}$$

follows by N2. For the second base case, let $\alpha = \triangle_i S_1$. Then $[\alpha] = \triangle_i[S_1]$ and $[\beta] \neq \triangle_i[S_1]$. If $\beta \neq \triangle_i S_2$ (for any S_2), then (C.11) follows by N4. If $\beta = \triangle_i S_2$ (for some S_2), then $[S_2] \neq [S_1]$. $d(S_1) \leq k \geq d(S_2)$, so $d(S_1 \sqcup S_2) \leq k$ and $\vdash \neg S_1 \doteq S_2$ holds by the induction hypothesis (in the "outmost" inductive proof). (C.11) follows by N3. Similarly, in the third base case when $\alpha = \triangledown_i S_1$ and $\beta \neq \triangle_i S_2$ (for any S_2) then (C.11) follows by N6, and if $\alpha = \triangledown_i S_1$ and $\beta = \triangledown_i S_2$, (C.11) follows by the induction hypothesis and N5.

Induction step (C.10) For the induction step, first let $\alpha = \neg\gamma$ – then $[\alpha] = \neg[\gamma]$. The induction hypothesis is that, for all γ', if $[\gamma] \neq [\gamma']$ then $\vdash \neg(\{\gamma\} \doteq \{\gamma'\})$. $[\beta] \neq \neg[\gamma]$. If β starts with a negation, $\beta = \neg\beta'$ where $[\gamma] \neq [\beta']$, then $\vdash \neg(\{\gamma\} \doteq \{\beta'\})$ by the induction hypothesis, and $\vdash \neg(\{\neg\gamma\} \doteq \{\neg\beta'\})$ by N7. If β does not start with a negation, (C.11) follows by N8. For the second part of the induction step, let $\alpha = (\gamma_1 \wedge \gamma_2)$. Then $[\alpha] = ([\gamma_1] \wedge [\gamma_2])$. $[\beta] \neq ([\gamma_1] \wedge [\gamma_2])$. If β is not a conjunction, then (C.11) follows by N10. If $\beta = (\beta_1 \wedge \beta_2)$ then $([\gamma_1] \wedge [\gamma_2]) \neq ([\beta_1] \wedge [\beta_2])$. Then either $[\gamma_1] \neq [\beta_1]$ or $[\gamma_2] \neq [\beta_2]$ and either $\vdash \neg(\{\gamma_1\} \doteq \{\beta_1\})$ or $\vdash \neg(\{\gamma_2\} \doteq \{\beta_2\})$ holds by the induction hypothesis, and then (C.11) follows by N9.

As in the base case, I use structural induction over T to prove

$$\text{If } d(T) \leq k+1 \text{ and } [T] = \{[\beta_1], \ldots, [\beta_l]\} \text{ then } \vdash T \doteq \{\beta_1, \ldots, \beta_l\} \tag{C.12}$$

Let $d(T) \leq k+1$ and $[T] = \{[\beta_1], \ldots, [\beta_l]\}$. Clearly, $d(\{\beta_j\}) \leq k+1$ $(1 \leq j \leq l)$.

Induction basis (C.12) For the base case, let $T = \{\alpha_1, \ldots, \alpha_l\}$ (in the base case T is a basic term, and it must consist of l formulae since there are l elements in $[T]$). $\{[\alpha_1], \ldots, [\alpha_l]\} = \{[\beta_1, \ldots, [\beta_l]\}$, and I assume that $[\alpha_j] = [\beta_j]$ $(1 \leq j \leq l)$ for simplicity (otherwise just change the indices). By T13 $\vdash T \doteq \{\alpha_1\} \sqcup \cdots \sqcup \{\alpha_l\}$, by (C.8) $\vdash \{\alpha_j\} \doteq \{\beta_j\}$, and by repeated applications of **Repl** and T3, $\vdash T \doteq \{\beta_1\} \sqcup \cdots \sqcup \{\beta_l\}$ and thus $\vdash T \doteq \{\beta_1, \ldots, \beta_l\}$ by T13 – which is what we needed to show for the basis in the structural induction over T.

Induction step (C.12) For the induction step, consider the case when $T = U \sqcup V$. In this case the proof of (C.12) is identical to the corresponding proof in the base case ($d(T \sqcup U) = 1$): Let $[U] = \{[\beta_1^U], \ldots, [\beta_{k_U}^U]\}$ and $[V] = \{[\beta_1^V], \ldots, [\beta_{k_V}^V]\}$, where each $\beta_j^U, \beta_j^V \in \{\beta_1, \ldots, \beta_l\}$.

1 $\vdash U \doteq \{\beta_1^U, \ldots, \beta_{k_U}^U\}$ Ind. hyp.

2 $\vdash V \doteq \{\beta_1^V, \ldots, \beta_{k_V}^V\}$ Ind. hyp.

3 $\vdash U \sqcup V \doteq \{\beta_1^U, \ldots, \beta_{k_U}^U\} \sqcup V$ **Repl**,1

4 $\vdash \{\beta_1^U, \ldots, \beta_{k_U}^U\} \sqcup V \doteq \{\beta_1^U, \ldots, \beta_{k_U}^U\} \sqcup \{\beta_1^V, \ldots, \beta_{k_V}^V\}$ **Repl**,2

5 $\vdash U \sqcup V \doteq \{\beta_1^U, \ldots, \beta_{k_U}^U\} \sqcup \{\beta_1^V, \ldots, \beta_{k_V}^V\}$ **MP**,T3,3,4

From several applications of T13, and **Repl**, we have that

6 $\vdash \{\beta_1^U, \ldots, \beta_{k_U}^U\} \sqcup \{\beta_1^V, \ldots, \beta_{k_V}^V\} \doteq$
$\{\beta_1^U, \ldots, \beta_{k_U}^U, \beta_1^V, \ldots, \beta_{k_V}^V\}$

7 $\vdash U \sqcup V \doteq \{\beta_1^U, \ldots, \beta_{k_U}^U, \beta_1^V, \ldots, \beta_{k_V}^V\}$ **MP**,T3,5,6

The sequence $\beta_1^U, \ldots, \beta_{k_U}^U, \beta_1^V, \ldots, \beta_{k_V}^V$ is a permutation, possibly with duplicates, of the sequence $\beta_1, \ldots, \beta_l$, so

$$\vdash U \sqcup V \doteq \{\beta_1, \ldots, \beta_l\}$$

by Lemma 4.2 and T3. For the second case in the induction step, let $T = U \sqcap V$ where $[U] = \{[\alpha_1^U], \ldots, [\alpha_{k_U}^U]\}$ and $[V] = \{[\alpha_1^V], \ldots, [\alpha_{k_V}^V]\}$.

Since $d(U) \leq k+1 \geq d(V)$, $d(\{\alpha^U_{j_U}\}) \leq k+1 \geq d(\{\alpha^V_{j_V}\})$ ($1 \leq j_U \leq k_U$, $1 \leq j_V \leq k_V$). I proceed exactly as in the base case ($d(T \sqcup U) = 1$) by first proving (C.4):

$$\vdash \{\alpha^U_1, \ldots, \alpha^U_{k_U}\} \sqcap \{\alpha^V_1, \ldots, \alpha^V_{k_V}\} \doteq \{\beta_1, \ldots, \beta_l\} \tag{C.13}$$

in the same way as in the base case. Again, we have that

$$\begin{aligned}
&\vdash \{\alpha^U_1, \ldots, \alpha^U_{k_U}\} \sqcap \{\alpha^V_1, \ldots, \alpha^V_{k_V}\} \doteq \\
&\quad (\{\alpha^U_1\} \sqcap \{\alpha^V_1\}) \sqcup (\{\alpha^U_1\} \sqcap \{\alpha^V_2\}) \sqcup \cdots \sqcup (\{\alpha^U_1\} \sqcap \{\alpha^V_{k_V}\}) \sqcup \\
&\quad (\{\alpha^U_2\} \sqcap \{\alpha^V_1\}) \sqcup (\{\alpha^U_2\} \sqcap \{\alpha^V_2\}) \sqcup \cdots \sqcup (\{\alpha^U_2\} \sqcap \{\alpha^V_{k_V}\}) \sqcup \\
&\quad \vdots \\
&(\{\alpha^U_{k_U}\} \sqcap \{\alpha^V_1\}) \sqcup (\{\alpha^U_{k_U}\} \sqcap \{\alpha^V_2\}) \sqcup \cdots \sqcup (\{\alpha^U_{k_U}\} \sqcap \{\alpha^V_{k_V}\})
\end{aligned} \tag{C.14}$$

by T13 and repeated applications of T12, **Repl** and the axioms of equality. The above expression contains exactly one term on the form $\{\alpha^U\} \sqcap \{\alpha^V\}$ for each pair $[\alpha^U] \in [U], [\alpha^V] \in [V]$. If $[\beta] \in [T]$, then $[\alpha^U_h] = [\beta] \in [U]$ and $[\alpha^V_j] = [\beta] \in [V]$, for some h, j. Since $d(\{\alpha^V_h\}), d(\{\alpha^V_j\}), d(\{\beta\}) \leq k+1$, $\vdash \{\alpha^V_j\} \doteq \{\beta\}$ and $\vdash \{\alpha^U_h\} \doteq \{\beta\}$ by (C.8) and $\vdash \{\alpha^U_h\} \sqcap \{\alpha^V_j\} \doteq \{\alpha^U_h\} \sqcap \{\beta\}$ by **Repl**. By T14 $\vdash \{\beta\} \sqcap \{\alpha^U_h\} \doteq \{\beta\}$, and thus $\vdash \{\alpha^U_h\} \sqcap \{\alpha^V_j\} \doteq \{\beta\}$, by equational calculus. If $[\alpha^U_h] \in [U]$ but $[\alpha^U_h] \notin [T]$, then, for every $[\alpha^V_j] \in [V]$, $[\alpha^U_h] \neq [\alpha^V_j]$, and, since $d(\{\alpha^U_h\}) \leq k+1 \geq d(\{\alpha^V_j\})$, $\vdash \neg(\{\alpha^U_h\} \doteq \{\alpha^V_j\})$ by (C.10) and thus $\vdash \{\alpha^U_h\} \sqcap \{\alpha^V_j\} \doteq \emptyset$ by T15. Similarly, if $[\alpha^V_j] \in [V]$ but $[\alpha^V_j] \notin [T]$, $\vdash \{\alpha^U_h\} \sqcap \{\alpha^V_j\} \doteq \emptyset$ for every $[\alpha^U_h] \in [U]$. Thus, (C.13) follows by (C.14), **Repl** and T13. I now use (C.13) to show the current induction step:

1	$\vdash U \doteq \{\alpha^U_1, \ldots, \alpha^U_{k_U}\}$	Ind. hyp.
2	$\vdash V \doteq \{\alpha^V_1, \ldots, \alpha^V_{k_V}\}$	Ind. hyp.
3	$\vdash U \sqcap V \doteq \{\alpha^U_1, \ldots, \alpha^U_{k_U}\} \sqcap \{\alpha^V_1, \ldots, \alpha^V_{k_V}\}$	**MP**,T5,1,2
4	$\vdash \{\alpha^U_1, \ldots, \alpha^U_{k_U}\} \sqcap \{\alpha^V_1, \ldots, \alpha^V_{k_V}\} \doteq \{\beta_1, \ldots, \beta_l\}$	(C.13)
5	$\vdash U \sqcap V \doteq \{\beta_1, \ldots, \beta_l\}$	**MP**,T3,3,4

This completes the proof of (C.12).

(C.12) can now be used show the the inductive step in the proof of the lemma, in exactly the same way I used (C.3) and (C.6) in the base case: Clearly, there exists α_j, β_j such that $[T_1] = \{[\alpha_1], \ldots, [\alpha_k]\}$ and $[T_2] = \{[\beta_1], \ldots, [\beta_m]\}$. First, consider the case that $[T_1] = [T_2]$:

1	$\vdash T_1 \doteq \{\alpha_1, \ldots, \alpha_k\}$	(C.12)
2	$\vdash T_2 \doteq \{\alpha_1, \ldots, \alpha_k\}$	(C.12)
3	$\vdash \{\alpha_1, \ldots, \alpha_k\} \doteq T_2$	**MP**,T2,2
4	$\vdash T_1 \doteq T_2$	**MP**,T3,1,3

Second, consider the case that $[T_1] \neq [T_2]$. As in the base case, I show that:

$$\vdash \neg(\{\alpha_1, \dots, \alpha_k\} \doteq \{\beta_1, \dots, \beta_m\}) \tag{C.15}$$

I assume that there is an $[\alpha_i] \in [T_1]$ such that $[\alpha_i] \notin [T_2]$ (the proof is equivalent in the case that $[\beta_i] \in [T_2]$ and $[\beta_i] \notin [T_1]$). Since $[\alpha_i] \neq [\beta_j]$ and $d(\{\alpha_i\}) \leq k+1 \geq d(\{\beta_j\})$ ($1 \leq j \leq m$), $\vdash \neg(\{\alpha_i\} \doteq \{\beta_j\})$ by (C.10) and $\vdash \neg((\{\alpha_i\} \doteq \{\beta_1\}) \vee \cdots \vee (\{\alpha_i\} \doteq \{\beta_m\}))$ by **Prop**. By **Prop** and N1, $\vdash \neg(\{\alpha_1, \dots, \alpha_k\} \preceq \{\beta_1, \dots, \beta_m\})$ and (C.15) follows by Lemma 4.1.12.

By (C.12) $\vdash \{\alpha_1, \dots, \alpha_k\} \doteq T_1$, and by (C.15), T3 and **Prop**

$$\vdash \neg(T_1 \doteq \{\beta_1, \dots, \beta_m\}) \tag{C.16}$$

By (C.12) $\vdash \{\beta_1, \dots, \beta_m\} \doteq T_2$, and by (C.16), T3, **Prop** and T2,

$$\vdash \neg T_1 \doteq T_2$$

Appendix D

Proof of Theorem 8.2

Theorem 8.2 is proved by proving the slightly different version in the following lemma.

Lemma D.1 Let

$$m = \begin{cases} \lfloor \frac{n}{3} \rfloor & n \text{ not divisible by } 3 \\ \frac{n}{3} - 1 & \text{otherwise} \end{cases}$$

Let

$$R \models \bigwedge_{i \in Ags} \overset{\rightsquigarrow}{\Diamond}_{ii}\{\frac{t}{u}\} \wedge \bigwedge_{i \neq j \in Ags} \overset{\rightsquigarrow}{\Diamond}_{ij} Oral(i,j)$$

(note that R is unique by Lemma 7.3.4). Let

$$R, (\vec{s}, \pi) \models \Gamma$$

There exists monotone $\vec{f}_{Ags} \in Str(Ags, R)$ such that for all $G \subseteq Ags$ with $|G| \geq n - m$, for all $\lambda \in out_R(\vec{f}_G, \vec{s})$ there exists a k such that

$$R, (\lambda[k], \pi) \models IC(G) \qquad \square$$

The Theorem follows from Lemma D.1 by taking $|G| > 2n/3$. It must be shown that $|G| \geq n - m$:

If $n|3$, then $m = n/3 - 1$, and $|G| \geq 2n/3 + 1$. $n - m = n - (n/3 - 1) = 2n/3 + 1 = |G|$.

If $n \nmid 3$, then $m = \lfloor n/3 \rfloor$, and $|G| \geq \lceil 2n/3 \rceil$. $n - m = n - \lfloor n/3 \rfloor = \lceil 2n/3 \rceil$, so $|G| \geq n - m$.

D.1 Proof of Lemma D.1

This proof is an adaption of a proof by Lamport, Shostak, & Pease (1982) and is, as noted previously, sketchy at some points.

Let

$$m = \begin{cases} \lfloor \frac{n}{3} \rfloor & n \text{ not divisible by } 3 \\ \frac{n}{3} - 1 & \text{otherwise} \end{cases}$$

$$R \models \bigwedge_{i \in Ags} \overset{\rightsquigarrow}{\Diamond}_{ii}\{\tfrac{t}{u}\} \wedge \bigwedge_{i \neq j \in Ags} \overset{\rightsquigarrow}{\Diamond}_{ij} Oral(i,j)$$

$$R, (\vec{s}, \pi) \models \Gamma$$

Clearly,

$$R_i(s) = \{(s_1, \ldots, s_n) : \\ s_i \in \wp^{fin}(OL), s_j \in \{\{\triangle_j\{\neg\neg \triangle_i \{\alpha\}\}, \{\triangle_j\{\neg \triangle_i \{\alpha\}\} : \alpha \in OL\}\} \quad (D.1)$$

for all $i \in Ags$ and $s \in \wp^{fin}(OL)$. Note that

$$n > 3m \quad (D.2)$$

because if $n|3$, $n > 3m = n - 3$; if $n \not| 3$, $n > 3\lfloor \frac{n}{3} \rfloor$ since $\lfloor \frac{n}{3} \rfloor < \frac{n}{3}$ when $n \not| 3$.

Before $\vec{f}_{Ags}$ is constructed, a few helpful notions are defined.

Given a set of agents $A \subseteq Ags$, the set of strings of m distinct agent names over A is:

$$AS(A,m) = \begin{cases} \{a_1 \cdots a_m : a_i \neq a_j, a_i, a_j \in A, i, j \in [1,m]\} & \text{if } m > 0 \\ \{\epsilon\} & \text{if } m = 0 \end{cases}$$

I write $AS(A)$ for the set $\cup_{1 \leq j \leq |A|} AS(A,j)$. I assume a fixed, but arbitrary, total ordering on $AS(Ags)$: $\gamma_1, \ldots, \gamma_p$. Henceforth, this ordering is assumed when comparing elements of $AS(Ags)$. Sometimes I will abuse notation and treat a member $\gamma \in AS(A)$ as a set.

Let $\gamma = a_1 \cdots a_k \in AS(Ags, k)$ and $\alpha \in AL$. I write

$$\triangle_\gamma \alpha$$

for

$$\triangle_{a_1}\{\triangle_{a_2} \cdots \{\triangle_{a_k}\{\alpha\}\}\}$$

If $\gamma = \epsilon$ (the empty string) then $\triangle_\gamma \phi = \phi$.

The following set of atomic propositions is used:

$$\Theta = \{c_i : i \geq 1\} \cup \{r_i : i \geq 0\} \cup \{sent_\gamma : \gamma \in AS(Ags)\} \\ \cup \{calc_\gamma : \gamma \in AS(Ags)\} \cup \{attack\}$$

The propositions are used by the agents to keep track of the progress of the algorithm. The strategy described below is effectively a linearized version of the recursive algorithm described by Lamport, Shostak, & Pease (1982), as is the following proofs. The reader must be warned that the idea behind the algorithm and the proofs may be hard obtain from the presentation below; the idea is however best explained by the mentioned recursive algorithm and corresponding proofs and I thus refer to Lamport, Shostak, & Pease (1982).

The λs correspond to the messages sent between the generals. The c_is and r_is are used to keep track of the recursion depth; specifically they respectively correspond to the different communication and reasoning steps. The $sent_\gamma$s and $calc_\gamma$s are used within respectively the communication and reasoning steps to keep track of which messages have been sent or calculated.

Now, $\vec{f}_{Ags} = \{f_i : i \in Ags\}$ is defined.

Let

$$f_i : \wp^{fin}(OL) \to (\wp^{fin}(OL))^n$$

be defined as follows for any agent $i \in Ags$:

$$f_i(s) = (s_1, \ldots, s_n)$$

where (explanation follows after the description)

step c_k**:** If $r_l \notin s$ for all l: Let $k = m$ if $c_m \in s$ or else the lowest number in $[0, m-1]$ such that $c_{k+1} \notin s$:

a) If $\{sent_\gamma : \gamma \in AS(Ags \setminus \{i\}, k)\} \nsubseteq s$: Let γ be the least member of $AS(Ags \setminus \{i\}, k)$ such that $sent_\gamma \notin s$.

i) If $\triangle_i\{\neg\neg \triangle_\gamma \{attack\}\} \in s$ and $\triangle_i \{\neg \triangle_\gamma \{attack\}\} \notin s$:

1) $s_i = s \cup \{\triangle_i\{\triangle_\gamma\{attack\}\}, sent_\gamma\}$

2) $j \in Ags \setminus (\{i\} \cup \gamma)$: $s_j = \{\triangle_j\{\neg\neg \triangle_i \{\triangle_\gamma\{attack\}\}\}\}$

3) $j \in \gamma$: $s_j = \emptyset$

ii) Otherwise:

1) $s_i = s \cup \{\neg \triangle_i \{\triangle_\gamma\{attack\}\}, sent_\gamma\}$

2) $j \in Ags \setminus (\{i\} \cup \gamma)$: $s_j = \{\triangle_j\{\neg \triangle_i \{\triangle_\gamma\{attack\}\}\}\}$

3) $j \in \gamma$: $s_j = \emptyset$

b) If $\{sent_\gamma : \gamma \in AS(Ags \setminus \{i\}, k)\} \subseteq s$:

i) $s_i = \begin{cases} s_i = s \cup \{c_{k+1}\} & \text{if } k < m \\ s_i = s \cup \{r_0\} & \text{if } k \geq m \end{cases}$

ii) $s_j, j \neq i$: $s_j = \emptyset$.

step r_0**:** If $r_0 \in s$ and $r_l \notin s$ for all $l \neq 0$:

a) s_i:

i) If $\{calc_\gamma : \gamma \in AS(Ags \setminus \{i\}, m+1)\} \nsubseteq s$: Let γ be the least member of $AS(Ags \setminus \{i\}, m+1)$ such that $calc_\gamma \notin s$. If $\triangle_i\{\neg\neg \triangle_\gamma \{attack\}\} \in s$ let $x = \triangle_\gamma\{attack\}$, otherwise let $x = \neg \triangle_\gamma \{attack\}$. Let $s_i = s \cup \{x, calc_\gamma\}$.

ii) If $\{calc_\gamma : \gamma \in AS(Ags \setminus \{i\}, m+1)\} \subseteq s$: $s_i = s \cup \{r_1\}$

b) $s_j, j \neq i$: $s_j = \emptyset$.

step r_k**:** If $r_k \in s, k > 0$, and $r_l \notin s$ for all $l > k$:

a) s_i:

i) If $\{calc_\gamma : \gamma \in AS(Ags \setminus \{i\}, m-k+1)\} \nsubseteq s$: Let γ be the least member of $AS(Ags \setminus \{i\}, m-k+1)$ such that $calc_\gamma \notin s$. Let

$$X = \{j : \triangle_j\{\triangle_\gamma\{attack\}\} \in s, j \in Ags \setminus \gamma\}$$

$$\overline{X} = \{j : \triangle_j\{\triangle_\gamma\{attack\}\} \notin s, j \in Ags \setminus \gamma\}$$

1) If $|X| > |\overline{X}|$: $s_i = s \cup \{\triangle_\gamma\{attack\}, calc_\gamma\}$

2) If $|X| \leq |\overline{X}|$: $s_i = s \cup \{\neg \triangle_\gamma \{attack\}, calc_\gamma\}$

ii) If $\{calc_\gamma : \gamma \in AS(Ags \setminus \{i\}, m-k+1)\} \subseteq s$:

1) If $k < m+1$: $s_i = s \cup \{r_{k+1}\}$.

2) If $k \geq m+1$: $s_i = s$.

b) $s_j, j \neq i$: $s_j = \emptyset$.

In this description of the function f_i, the notation **step..:** and the different numberings a),i),1), etc., are just tags to be able to refer to the different parts of the definition later. It is clear from this description that each f_i is well-defined; for each s it defines s_i and s_j for each j, and all the different cases are mutually exclusive. It is also clear that $f_i(s) \subseteq R_i(s)$ for all $i \in Ags$ and $s \in \wp^{fin}(OL)$ (see eq. (D.1)), and thus $\vec{f}_{Ags} \in Str(Ags, R)$, and that $\vec{f}_{Ags}$ are monotone.

Let $G \subseteq Ags$ such that

$$|G| \geq n - m \tag{D.3}$$

and let $\lambda \in out_R(\vec{f}_G, \vec{s})$. If

$$\exists_k \text{ such that } R, (\lambda[k], \pi) \models IC(G) \tag{D.4}$$

then the Lemma is true. Thus the rest of the proof is a proof of eq. (D.4).

Note that, since $\vec{f}_G$ is monotone, $\lambda[k]^i \subseteq \lambda[k+1]^i$ for any $i \in G$ and any $k \geq 0$.

It is easy to see that any agent i who uses the strategy f_i, from the start position $\vec{s}$, will use all of the following parts of the function, in the following order:

$$\text{step}c_0, \ldots, \text{step}c_m, \text{step}r_0, \ldots, \text{step}r_{m+1}$$

The role of the c_k and r_k propositions is exclusively to keep track of this progress. Within each step agent i will do a number[1] different substeps. For the c_k steps these substeps, a number of uses of step c_ka) and one use of step c_kb), involve sending and receiving messages and their progress is controlled by the $sent_\gamma$ propositions; for the r_k steps the substeps, a number of uses of step r_ka)i) and one use of step $r_k a)ii)$, involve calculating majorities and their progress is controlled by the $calc_\gamma$ propositions. All substeps in a step will eventually be used, and the agent goes on to the next step as outlined above, except for step r_{m+1} where the function will "terminate".

Note also that each agent in G will execute these step simultaneously; the strategy function, which every agent in G uses, defines a fixed number of times the function is used between each step. This number does not depend on e.g. the messages the agent receives. Thus, each agent will go from one step to the next at exactly the same time. Particularly, there exists k_p, $p \in [0, m+1]$, such that

$$r_p \in \lambda[k_p]^i \text{ and } r_p \notin \lambda[k_p - 1]^i$$

for each $i \in G$. These notions will be used in the proofs below.

The following Lemma D.3 is used to prove eq. (D.4), and uses an intermediate result in Lemma D.2. Informally, Lemma D.2 solves the case when the commanding general is loyal.

Lemma D.2 For every $m' \in [0, m]$, k s.t. $n > 2k + m$, $Ags' \subseteq Ags$ s.t. $|Ags'| = n - m + m'$, $G' \subseteq Ags' \cap G$ s.t. $|G'| \geq |Ags'| - k$, $\gamma \in AS(Ags \setminus Ags', m - m')$, $l \in G'$:

$$R, (\lambda[k_{m'+1}, \pi)] \models IC(G', \triangle_l\{\triangle_\gamma\{attack\})\qquad\square$$

[1]Actually $|AS(Ags \setminus \{i\}, k)| + 1$ for the c_k steps and $|AS(Ags \setminus \{i\}, m - k + 1)|$ for the r_i steps.

PROOF The proof is by induction over m'.

For the base case, $m' = 0$ and $|\gamma| = m$. Assume first that agent l knows $\triangle_l\{\neg\neg \triangle_\gamma \mathit{attack}\}$ and does not know $\triangle_l\{\neg \triangle_\gamma \mathit{attack}\}$ right before step c_m. In step c_ma)i), since $\gamma \in AS(Ags \setminus \{l\}, m)$, l will send $\triangle_l\{\triangle_\gamma \mathit{attack}\}$ to himself, and $\triangle_j\{\neg\neg \triangle_l \{\triangle_\gamma \mathit{attack}\}\}$ to every $j \in G' \setminus \{l\} \subseteq Ags \setminus (\{l\} \cup \gamma)$ and, by monotonicity, $\triangle_j\{\neg\neg \triangle_l \{\triangle_\gamma \mathit{attack}\}\} \in \lambda[k_0]^j$. Let $j \in G' \setminus \{l\}$. Since $j \in G$, j uses strategy f_j and will, since $l\gamma \in AS(Ags \setminus \{j\}, m+1)$ send $x = \triangle_l\{\triangle_\gamma \mathit{attack}\}$ to himself in step r_0a)i). Thus, $\triangle_l\{\triangle_\gamma \mathit{attack}\} \in \lambda[k_1]^j$ for any $j \in G' \setminus \{l\}$. It can be shown that $\neg \triangle_l \{\triangle_\gamma \mathit{attack}\} \notin \lambda[k_1]^j$: there is no such formula in $\lambda[0]^j$ (which is described by Γ; $l \neq j$) and since it is impossible for another agent to communicate such a formula to j (restriction on messages) it must have been inferred by j in step r_0a)i) – which is impossible. Also, by monotonicity ($l \in G$), $\triangle_l\{\triangle_\gamma \mathit{attack}\} \in \lambda[k_1]^l$. It can be shown that $\neg \triangle_l \{\triangle_\gamma \mathit{attack}\} \notin \lambda[k_1]^l$: it cannot have been communicated, it was not inferred by l in step c_ma)i), and the only possibility for $\neg \triangle_l \{\triangle_\gamma \mathit{attack}\} \in \lambda[0]^l$ is that $l = cg$ and $\gamma = \epsilon$ ($m = 0$) — but that is also impossible because by the assumption at the beginning of the base case $\triangle_{cg}\{\neg \mathit{attack}\} \notin \lambda[0]^{cg}$ which implies that $\neg \triangle_{cg} \{\mathit{attack}\} \notin \lambda[0]^{cg}$ by construction of Γ. Thus, $R, (\lambda[k_1], \pi) \models IC(G', \triangle_l\{\triangle_\gamma \mathit{attack}\})$. The opposite assumption, that right before step c_m either agent l does not know $\triangle_l\{\neg\neg \triangle_\gamma \mathit{attack}\}$ or agent l knows both $\triangle_l\{\neg\neg \triangle_\gamma \mathit{attack}\}$ and $\triangle_l\{\neg \triangle_\gamma \mathit{attack}\}$, leads to the fact that l will use case ii) in step c_ma) and that $\neg \triangle_l \{\triangle_\gamma \mathit{attack}\} \in \lambda[k_1]^j$ and $\triangle_l\{\triangle_\gamma \mathit{attack}\} \notin \lambda[k_1]^j$ for each $j \in G'$ by a symmetrical argument[2] and thus $(R, \lambda[k_1], \pi) \models IC(G', \triangle_l\{\triangle_\gamma \mathit{attack}\})$ also in this case.

For the inductive step, let $m' > 1$ and assume that the lemma holds for $m' - 1$. Let k, Ags', G', γ and l be as described in the lemma. In step $c_{m-m'}$a) l sends either $\triangle_j\{\neg\neg \triangle_l \{\triangle_\gamma \mathit{attack}\}\}$ (case $c_{m-m'}$a)i)) or $\triangle_j\{\neg \triangle_l \{\triangle_\gamma \mathit{attack}\}\}$ (case $c_{m-m'}$a)ii)) to each $j \in Ags \setminus (\{l\} \cup \gamma)$, where $|\gamma| = m - m'$. Assume the former. Then, in step $c_{m-m'}$a), l sends $\triangle_l\{\triangle_\gamma \mathit{attack}\}$ to himself. In step $c_{m-m'+1}$a), each $j \in Ags \setminus (\gamma \cup \{l\})$ decides $\triangle_j\{\triangle_l\{\triangle_\gamma \mathit{attack}\}\}$ and not $\neg \triangle_j \{\triangle_l\{\triangle_\gamma \mathit{attack}\}\}$, so $\triangle_j\{\triangle_l\{\triangle_\gamma \mathit{attack}\}\} \in \lambda[k_{m'}]^j$ by monotonicity and $\neg \triangle_j \{\triangle_l\{\triangle_\gamma \mathit{attack}\}\} \notin \lambda[k_{m'}]^j$ by a similar argument to the base case: the only possibility would be that $\neg \triangle_j \{\triangle_l\{\triangle_\gamma \mathit{attack}\}\} \in \lambda[0]^j$ which is impossible since by construction of Γ there is no such formula in $\lambda[0]^j$ (even if $\gamma = \epsilon$). Let $Ags'' = Ags' \setminus \{l\}$, $G'' = G' \setminus \{l\}$ and $\gamma' = l\gamma$. $|Ags''| = |Ags'| - 1 = n - m + m' - 1$. $G'' \subseteq Ags'' \cap G$, and $|G''| = |G'| - 1 \geq |Ags'| - k - 1 = |Ags''| - k$. $\gamma \in AS(Ags \setminus Ags'', m - (m' - 1))$. Let $j \in G''$:

$$R, (\lambda[k_{m'}], \pi) \models IC(G' \setminus \{l\}, \triangle_j\{\triangle_l\{\triangle_\gamma \mathit{attack}\}\})$$

by the induction hypothesis. Since $j \in G''$, $j \in Ags \setminus (\gamma \cup \{l\})$, so $\triangle_j\{\triangle_l\{\triangle_\gamma \mathit{attack}\}\} \in \lambda[k_{m'}]^j$. By interactive consistency, $\triangle_j\{\triangle_l\{\triangle_\gamma \mathit{attack}\}\} \in \lambda[k_{m'}]^i$ and $\neg \triangle_j \{\triangle_l\{\triangle_\gamma \mathit{attack}\}\} \notin \lambda[k_{m'}]^i$ for every $i \in G' \setminus \{l\}$. This holds for every $j \in G''$.

[2]Again, the only possibility of $\triangle_l\{\triangle_\gamma \mathit{attack}\} \in \lambda[k_1]^j$ would be that $l = j = cg$,$m = 0$ and $\gamma = \epsilon$ and that $\triangle_{cg}\mathit{attack} \in \lambda[0]^{cg}$. If the first case in the opposite assumption were true, by construction of Γ $\triangle_{cg}\{\mathit{attack}\} \notin \lambda[0]^{cg}$ since $\triangle_{cg}\{\neg\neg \mathit{attack}\} \notin \lambda[0]^{cg}$. The second case cannot be true, because if cg knows $\triangle_{cg}\{\neg\neg \mathit{attack}\}$ and $\triangle_{cg}\{\neg \mathit{attack}\}$ before step c_0 then $\triangle_{cg}\{\neg\neg \mathit{attack}\}, \triangle_{cg}\{\neg \mathit{attack}\} \in \lambda[0]^{cg}$ which is impossible by construction of Γ.

In other words:

For every $i \in G' \setminus \{l\}$: for every $j \in G''$: $\triangle_j \{\triangle_l\{\triangle_\gamma attack\}\} \in \lambda[k_{m'}]^i$

Let $i \in G' \setminus \{l\}$. In step $r_{m'}$ agent i has $|\overline{X}| = |(Ags \setminus (\gamma \cup \{l\})) \setminus (G' \setminus \{l\})| = n - (m - m' + 1) - (|G'| - 1) \leq n - m + m' - (n - m + m' - k) = k$ (because $|G'| \geq n - m + m' - k$). $|X| + |\overline{X}| = |Ags \setminus (\gamma \cup \{l\})| = n - m + m' - 1 > 2k + m - m + m' - 1 \geq 2k$ (because $n > 2k + m$ and $m' > 0$). Thus, since $|\overline{X}| \leq k$ and $|X| + |\overline{X}| > 2k$, $|X| > |\overline{X}|$ and $\triangle_l\{\triangle_\gamma attack\} \in \lambda[k_{m'+1}]^i$ because i sends $\triangle_l\{\triangle_\gamma attack\}$ to himself in step $r_{m'}$a)i)1). It can be shown that $\neg \triangle_l \{\triangle_\gamma attack\} \notin \lambda[k_{m'+1}]^i$ (the only possibility of the contrary is that $|\overline{X}| \geq |X|$; $\neg \triangle_l \{\triangle_\gamma attack\} \notin \lambda[0]^i$ since $l \neq i$). This holds for every $i \in G' \setminus \{l\}$. It can be shown that $\neg \triangle_l \{\triangle_\gamma attack \notin \lambda[k_{m'+1}]^l$: the only possibility of $\neg \triangle_l \{\triangle_\gamma attack \notin \lambda[0]^l$ is that $l = cg$ and $\gamma = \epsilon$ but since cg sends $\triangle_j\{\neg\neg \triangle_{cg} \{attack\}\}$ in step $c_{m-m'}$ $\triangle_{cg}\{\neg attack\} \notin$
$lambda[0]^{cg}$ and thus $\neg \triangle_{cg} \{attack\} \notin$
$lambda[0]^{cg}$ by construction of Γ; the only other possibility would be that the formula was inferred by l in step $c_{m-m'}$ which it was not. Thus, since also $\triangle_l\{\triangle_\gamma attack\} \in \lambda[k_{m'+1}]^l$,

$$R, (\lambda[k_{m'+1}], \pi) \models IC(G', \triangle_l\{\triangle_\gamma attack\})$$

The opposite assumption, that l sends $\triangle_j\{\neg \triangle_l \{\triangle_\gamma attack\}\}$ to each $j \in Ags \setminus (\{l\} \cup \gamma)$, leads to the fact that $\neg \triangle_l \{\triangle_\gamma attack\} \in \lambda[k_{m'+1}]^i$ and $\triangle_l\{\triangle_\gamma attack\} \notin \lambda[k_{m'+1}]^i$ for every $i \in G'$ by a symmetrical argument, so

$$R, (\lambda[k_{m'+1}], \pi) \models IC(G', \triangle_l\{\triangle_\gamma attack\})$$

holds also in this case. This completes the inductive step. ■

Lemma D.3 For every $m' \in [0, m]$, for every $Ags' \subseteq Ags$ such that $|Ags'| = n - m + m'$, for every $G' \subseteq Ags' \cap G$ such that $|G'| \geq n - m$, for every $\gamma \in AS(Ags \setminus Ags', m - m')$, for every $l \in Ags'$

$$R, (\lambda[k_{m'+1}], \pi) \models IC(G', \triangle_l\{\triangle_\gamma attack\}) \quad \text{(D.5)}$$

□

PROOF The proof is by induction over m'.

For the base case, let $m' = 0$. Since $G' \subseteq Ags'$ and $|G'| \geq n - m = |Ags'|$, $G' = Ags'$ and thus $l \in G'$. Eq. (D.5) follows from Lemma D.2 by taking $k = 0$.

For the inductive step, assume that the lemma holds for $m' - 1$. First, consider the case when $l \in G'$. Then, (D.5) again follows from Lemma D.2 by taking and $k = m'$, since $n > 3m \geq 2m' + m$ (eq. (D.2) and $m' \leq m$).

Second, consider the case when $l \notin G'$. Let $Ags' \subseteq Ags$, $|Ags'| = n - m + m'$, $G' \subseteq Ags' \cap G$, $|G'| \geq n - m$, $\gamma \in AS(Ags \setminus Ags', m - m')$ and $l \in Ags'$. Let $Ags'' = Ags' \setminus \{l\}$. Ags'' can be used with the induction hypothesis as follows. $|Ags''| = |Ags'| - 1 = n - m + m' - 1$, $G' \subseteq Ags'' \cap G$, $\gamma' = l\gamma \in AS(Ags \setminus Ags'', m - (m' - 1))$. By the induction hypothesis, for every $j \in Ags''$

$$R, (\lambda[k_{m'}], \pi) \models IC(G', \triangle_j\{\triangle_l\{\triangle_\gamma attack\}\})$$

Note that, if γ is seen as a set, $\gamma = Ags \setminus Ags'$ since $\gamma \subseteq Ags \setminus Ags'$ and $|\gamma| = m - m' = |Ags \setminus Ags'|$. Thus, $Ags'' = Ags \setminus (\gamma \cup \{l\})$. By the induction hypothesis, in step $r_{m'}$, for each $j \in Ags'' = Ags \setminus (\gamma \cup \{l\})$ either all agents in G' know $\triangle_j\{\triangle_l\{\triangle_\gamma attack\}\}$ or all agents in G' does not know $\triangle_j\{\triangle_l\{\triangle_\gamma attack\}\}$. Consider step $r_{m'}$a)i) for each $i \in G'$: i must calculate the values X and $\overline{X}$ for $l\gamma \in AS(Ags \setminus \{i\}, m - m' + 1)$. Every agent $i \in G$ will calculate the values for will get the *same values* for the sets X and $\overline{X}$, and thus will either all conclude that $\triangle_l\{\triangle_\gamma attack\}$ or all conclude that $\neg \triangle_l \{\triangle_\gamma attack\}$ (but not both) in step $r_{m'}$a)i). It is also easy to see that the agents in G' could not have come to know either $\triangle_l\{\triangle_\gamma attack\}$ or $\neg \triangle_l \{\triangle_\gamma attack\}$ by *reasoning* before step $r_{m'}$, and neither by *communication* before or in step $r_{m'}$ — the latter since $l \notin G'$ and no agents can send a formula on one of the two forms to an agent $i \neq l$, nor is $\triangle_l\{\triangle_\gamma attack\} \in \lambda[0]^j$ for any $j \in G'$ by construction of Γ (since $l \notin \square'$). Thus, when round $r_{m'}$ is finished G' have interactive consistency:

$$R, (\lambda[k_{m'+1}], \pi) \models IC(G', \triangle_l\{\triangle_\gamma attack\})$$

This completes the inductive step, and the proof. ■

(D.4) follows from Lemma D.3 by taking $m' = m$, $Ags' = Ags$ ($|Ags'| = n = n - m + m'$), $G' = G$ ($|G| = n - m$ by eq. (D.3)), $\gamma = \epsilon$ and $l = cg$. This completes the proof of (D.4), and thus the Lemma.