Exploring Logical Perspectives on Distributed Information and its Dynamics

MSc Thesis

written by

Floris Roelofsen
(born July 25, 1980 in Amsterdam, the Netherlands)

under the supervision of **Prof Dr Johan van Benthem**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *University of Amsterdam*.

Date of the public defense:
August 29, 2005

Members of the Thesis Committee:
Prof Dr Johan van Benthem
Prof Dr Jan van Eijck
Dr Hans van Ditmarsch
Dr Peter van Emde Boas
Dr Maricarmen Martinez

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Contents

Acknowledgments

At the 2004 European Summer School for Logic, Language, and Information in Nancy I first met Johan van Benthem. His presentation was magical, by far the most impressive scientific presentation I had ever attended. Two weeks later I started the Master of Logic programme in Amsterdam, and asked Johan to supervise my thesis work. I am very grateful for his agreeing to do so, and his very instructive and enjoyable guidance along the way.

I have worked on two different topics this year. First, I became interested in non-monotonic reasoning and semantics based on minimal models. To get to grips with these ideas, I studied their possible applications to a framework I was already familiar with, namely, that of multi-context systems. I am very grateful to Michiel van Lambalgen for his initial encouragements, to Luciano Serafini and Gerhard Brewka for comments on my writings, and to the University of Amsterdam for sponsoring my visit to Trento in January, during which most of the work reported here was accomplished.

When I started working on dynamic epistemic logic in February, I hardly knew anything about it. Discussions with Michael Franke, Paul Harrenstein, Fenrong Liu, Maricarmen Martinez, Siewert van Otterloo, Olivier Roy, and especially with Yanjing Wang helped me on my way. Later on, Hans van Ditmarsch, Jan van Eijck, Jelle Gerbrandy, Wiebe van der Hoek, Barteld Kooi, Tomasz Sadzik, and Josh Snyder provided very useful feedback. David Pearce very attentively arranged my stay in Madrid in April and May.

Parts of this thesis will be presented at various conferences [29, 30, 32]. I would like to thank the ILLC and IJCAI for generously providing the scholarships necessary to attend these events, and Johan van Benthem, Dick de Jongh, and Benedikt Löwe for their help in obtaining these scholarships.

On a more personal note, it was a great pleasure to spend a year “back home”. I especially enjoyed spending time with my parents and sister again. I found a great dance partner, new friends, and... lost my heart. Gracias!

Overview

This thesis is divided into two parts. Each part explores a particular logical perspective on distributed information and its dynamics.

The first part takes the perspective of epistemic and dynamic epistemic logic. We first explain why distributed knowledge does not always comply with the principle of full communication, as observed by van der Hoek, van Linder, and Meyer [43] and Gerbrandy [16], and give a complete characterization of the class of models in which distributed knowledge *does* comply with the principle of full communication. Then, we address an issue raised by van Benthem [40]: is there a natural extended notion of bisimulation that matches the expressive power of distributed knowledge operators in epistemic logic. Next, we show that distributed knowledge operators (or more generally, intersection modalities) can be incorporated into the logic of communication and change, recently proposed by van Benthem, van Eijck, and Kooi [42]. At last, we discuss a more conceptual point. Distributed knowledge is what can be established by a group of agents through a particular kind of communication: everyone writes down everything he knows, all that is put together, and what follows from the accumulated facts is distributed knowledge. Other, arguably more interactive kinds of communication may be considered as well. Van Benthem [39], for example, is interested in the information that can be established by a group of agents through *public* announcements. We discuss the information that can be established by a group of agents given a certain *communication network*.

The second part of the thesis takes the perspective of multi-context systems. These systems describe the distribution and flow of information among a number of contexts (people, databases, etc.). We provide a simplified semantics for the basic systems and define generalizations of this semantics for various extended systems.

Part I

Epistemic Logic

Chapter 1

Expressive Power and Full Communication

Distributed knowledge is a standard notion in epistemic logic [13, 23]. Intuitively, a formula φ is distributed knowledge among a group of agents B iff φ follows from the knowledge of all individual agents in B put together. Semantically, φ is distributed knowledge among B iff φ is true in all worlds that *every* agent in B considers possible. This chapter addresses two issues concerning distributed knowledge.

Full communication. Van der Hoek, van Linder, and Meyer [43] argued that, to be of any use at all, a notion of group knowledge should comply with what they call the *principle of full communication*: whenever φ is considered group knowledge, it should be possible for the members of the group to establish φ through communication (this will be made more precise below). Van der Hoek, van Linder, and Meyer [43] and Gerbrandy [16] showed that distributed knowledge does not generally comply with the principle of full communication, but does in certain special model classes. It is not clear, however, *why* distributed knowledge does not generally comply with the principle of full communication, and why it does in these special model classes. Moreover, it is not known whether these model classes are *complete*, that is, whether they comprise all models in which distributed knowledge complies with the principle of full communication. We will provide a simple analysis of the problem and a complete characterization of the class of models in which distributed knowledge complies with the principle of full communication.

Expressive power. A standard notion of structural equivalence between epistemic models is that of *bisimilarity*. This notion (to be defined below) perfectly matches the expressive power of basic epistemic formulas (formulas without distributed knowledge operators): if two models are bisimilar, then they satisfy exactly the same basic formulas. But adding distributed knowledge to the basic language yields a more expressive language, whose formulas may be able to distinguish bisimilar models. Is there a natural extended notion of bisimulation that matches the expressive power of the language with distributed knowledge? This question is the first of a list of open problems in a recent survey by van Benthem [40]. We will define and analyze a suitable extended notion of bisimulation, corresponding model comparison games, and a closely related extended notion of modal saturation.

The chapter is organized as follows. Section 1.1 reviews some basic notions from epistemic logic. Section 1.2 is concerned with the extent to which distributed knowledge complies with the principle of full communication, and section 1.3 introduces notions of bisimulation and saturation, as well as related model comparison games, to capture the expressive power of distributed knowledge operators. Sections 1.2 and 1.3 each conclude with a short summary and pointers to related work.

1.1 Epistemic Logic

The following notions are all standard in epistemic logic [13, 23]. A countable set of proposition letters $\mathcal{P}$ and a finite set of agents $\mathcal{A}$ is assumed to be given throughout our general discussion and clear from the context in particular examples.

Languages. The basic epistemic language consists of all formulas that can be built from proposition letters in $\mathcal{P}$ using conjunction, negation, and a modal operator K_a for every agent $a \in \mathcal{A}$. $K_a\varphi$ stands for *agent a knows that φ is true.* The basic epistemic language is denoted by $\mathcal{L}_K$:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \psi \mid K_a\varphi$$

One standard way to extend the basic language is to add a modal operator D_B for every group of agents $B \subseteq \mathcal{A}$. $D_B\varphi$ stands for *φ is distributed knowledge among B*. The resulting language is denoted by $\mathcal{L}_D$:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \psi \mid K_a\varphi \mid D_B\varphi$$

Models. A model M is a triple (W, R, V), where:

- W is a non-empty set of worlds,
- $R : \mathcal{A} \to \wp(W \times W)$
- $V : W \to \wp(\mathcal{P})$

R assigns to every agent $a \in \mathcal{A}$ a so-called *accessibility relation* on W. Intuitively, $(w, v) \in R(a)$ means that in world w, agent a considers world v possible. Accessibility relations are often assumed to be equivalence relations, or to have other less restrictive properties, but for sake of generality we do not commit ourselves to any such specific assumptions here. V associates every world $w \in W$ with a subset of $\mathcal{P}$, the proposition letters that are true in w. If $M = (W, R, V)$ is a model and w is a particular world in W, then (M, w) is called a pointed model, and w is called its actual world. We will often simply refer to pointed models as models. The *information state* $[M, w]_a$ of an agent a in (M, w) is the set of worlds that a considers possible in (M, w). Similarly, the information state $[M, w]_B$ of a group of agents B in (M, w) is the set of worlds that every agent $a \in B$ considers possible in (M, w):

$$\begin{array}{lcl} [M, w]_a & = & \{v \in W \mid (w, v) \in R_a\} \\ [M, w]_B & = & \{v \in W \mid (w, v) \in R_a \text{ for all } a \in B\} \end{array}$$

Semantics. The satisfaction relation $\vDash$ between pointed models and formulas in $\mathcal{L}_K$ or $\mathcal{L}_D$ is recursively defined as follows:

$$\begin{array}{lll} M, w \vDash p & \text{iff} & p \in V(w) \\ M, w \vDash \neg\varphi & \text{iff} & M, w \nvDash \varphi \\ M, w \vDash \varphi \wedge \psi & \text{iff} & M, w \vDash \varphi \text{ and } M, w \vDash \psi \\ M, w \vDash K_a\varphi & \text{iff} & M, v \vDash \varphi \text{ for all } v \in [M, w]_a \\ M, w \vDash D_B\varphi & \text{iff} & M, v \vDash \varphi \text{ for all } v \in [M, w]_B \end{array}$$

Intuitively, the $K_a\varphi$ clause says that an agent knows φ to be true just in case φ is true in all worlds she considers possible. Similarly, the $D_B\varphi$ clause says that φ is distributed knowledge among B just in case φ is true in all worlds that *every* agent in B considers possible. A set of formulas Φ *entails* a formula φ, $\Phi \Vdash \varphi$, iff every pointed model that satisfies all formulas in Φ also satisfies φ. A set of formulas Φ is *consistent* or *satisfiable* iff there is a pointed model that satisfies all formulas in Φ. One set of formula Φ is consistent with another set of formulas Σ iff $\Phi \cup \Sigma$ is consistent. A set of formulas Φ is satisfiable in an information state iff that information state contains a world that satisfies all formulas in Φ. The *theory* of a world in a model is the set of all formulas true in that world. A world is consistent with a set of formulas Φ iff its theory is consistent with Φ.

1.2 Full Communication

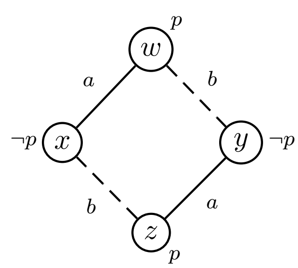
Van der Hoek, van Linder, and Meyer [43] argue that, to be of any use at all, a notion of group knowledge should comply with what they call the *principle of full communication*: whenever φ is considered group knowledge, it should be possible for the members of the group to establish φ through communication. To make this more precise, they define the *knowledge set* of a group of agents B in a model (M, w) to be the set of all $\mathcal{L}_K$-formulas that at least one agent in B knows to be true in (M, w):

$$\mathrm{KS}_B(M, w) = \{\psi \in \mathcal{L}_K \mid M, w \vDash K_a\psi \text{ for some } a \in B\}$$

Then they take it that a formula can be established through communication by a group of agents iff that formula is entailed by the knowledge set of that group. So distributed knowledge complies with the principle of full communication iff for all $\varphi \in \mathcal{L}_K$:

$$M, w \vDash D_B\varphi \quad \Rightarrow \quad \mathrm{KS}_B(M, w) \Vdash \varphi \tag{1.1}$$

Van der Hoek, van Linder, and Meyer [43] as well as Gerbrandy [16] show that (1.1) does *not* generally hold. To see this, consider the model depicted below (all accessibility relations are equivalence relations here). Take $B = \{a, b\}$. Then, p is distributed knowledge among B in w, but p is not entailed by B's knowledge set in w.



In [43] and [16] various classes of models are identified in which distributed knowledge *does* comply with the principle of full communication. Van der Hoek, van Linder, and Meyer define a model M to be *distinguishing* iff for every world w in M there is an $\mathcal{L}_K$-formula that is true in w and nowhere else M. They show that distributed knowledge complies with the principle of full communication in finite, distinguishing models. Gerbrandy generalizes this result. He defines M to be *locally distinguishing* iff for every w in M, every $v \in \bigcup_{a\in\mathcal{A}}[M, w]_a$, and every $a \in \mathcal{A}$, there is an $\mathcal{L}_K$-formula

φ_a such that $M, v \vDash \varphi_a$ iff $v \in [M, w]_a$. He shows that distributed knowledge complies with the principle of full communication in locally distinguishing models. Gerbrandy also defines a model M to be *full* iff for every group $B \subseteq \mathcal{A}$ and every $w \in W$, every set of $\mathcal{L}_D$-formulas that is consistent with $\mathrm{KS}_B(M, w)$ is satisfiable in $[M, w]_B$. He proves that distributed knowledge complies with the principle of full communication in full models.

It is not clear, however, *why* distributed knowledge does not generally comply with the principle of full communication, and why it does in models that are finite, (locally) distinguishing and/or full. Moreover, it is not clear whether *all* models in which distributed knowledge complies with the principle of full communication are finite, (locally) distinguishing and/or full. And if not, whether a complete characterization of such models can be given.

Section 1.2.1 gives a simple analysis of why distributed knowledge does not always comply with the principle of full communication. This analysis yields a rather general class of models in which distributed knowledge *does* comply with the principle. Section 1.2.2 relates the defining properties of this class, called *tightness* and *epistemic saturation*, with notions familiar from modal logic. In particular, epistemic saturation is shown to be equivalent with modal saturation. In section 1.2.3 we compare our model class with the model class defined by van der Hoek, van Linder, and Meyer. We show that every finite, distinguishing model is tight and saturated, and moreover, that there is an interesting intermediate model class consisting of generated submodels of the canonical model. In section 1.2.4 we show that our model class can be generalized in various ways, which finally leads to a complete characterization of models in which distributed knowledge complies with the principle of full communication. Interestingly, the characteristic property of such models turns out to be a weak version of Gerbrandy's fullness property.

1.2.1 Distributed Knowledge and Full Communication

Why does distributed knowledge not comply with the principle of communication? One answer to this question is that in general *an agent's information state may not contain every world that is consistent with her knowledge set.* Consider, for example, the model depicted above: everything a knows in w is true in y. So y is consistent with a's knowledge set in w. Still, y does not belong to a's information state in w. Similarly, x does not belong to b's information state in w, even though everything b knows in w is true in y, that is, y is consistent with b's knowledge set in w. As a consequence, the intersection of two information states may sometimes yield more information than the union of the corresponding knowledge sets. To continue the above example, a and b's knowledge sets in w are identical. Thus, taking their union does not yield any new information. However, a and b's information states in w are different, and their intersection yields a new, more informative state. As a result, p is distributed knowledge among a and b in w, even though it is not entailed by the union of their knowledge sets.

This explanation, trivial as it may seem, leads us to the characterization of a rather general class of models in which distributed knowledge *does* comply with the principle of full communication, namely, the class of models in which every agent's information state *does* contain every world that is consistent with her knowledge set.

This requirement can be split into two sub-requirements. First, every set of formulas that is consistent with an agent's knowledge set must be satisfiable in the agent's information state. We call this requirement *epistemic saturation*.

Definition 1.1 (Epistemic Saturation) *A model $M = (W, R, V)$ is* epistemically saturated *iff for all $w \in W$ and all $a \in \mathcal{A}$, every set of $\mathcal{L}_K$-formulas that is consistent with $KS_a(M, w)$ is satisfiable in $[M, w]_a$.*

Second, every world that is consistent with an agent's knowledge set must be contained in the agent's information state. We call this requirement *tightness*.

Definition 1.2 (Tightness) *A model $M = (W, R, V)$ is* tight *iff for all $w \in W$ and all $a \in \mathcal{A}$, every world in M that is consistent with $KS_a(M, w)$ is in $[M, w]_a$.*

Notice that epistemic saturation and tightness are complementary requirements: the former requires that a model contains *enough worlds*; the latter requires that a model provides for *enough access*.

We now show that distributed knowledge complies with the principle of full communication in tight and epistemically saturated models.

Proposition 1.3 *In tight and epistemically saturated models distributed knowledge complies with the principle of full communication.*

Proof. Let $M = (W, R, V)$ be tight and epistemically saturated. To prove:

$$M, w \vDash D_B\varphi \quad \Rightarrow \quad \mathrm{KS}_B(M, w) \Vdash \varphi$$

Suppose that $\mathrm{KS}_B(M, w) \nvdash \varphi$. Then $\{\neg\varphi\} \cup \mathrm{KS}_B(M, w)$ is consistent. Let $a \in B$. Then $\{\neg\varphi\} \cup \mathrm{KS}_B(M, w)$ is consistent with $\mathrm{KS}_a(M, w)$ so, by epistemic saturation of M, $[M, w]_a$ contains a world w_a satisfying all formulas in $\{\neg\varphi\} \cup \mathrm{KS}_B(M, w)$. Thus, w_a is consistent with $\mathrm{KS}_b(M, w)$ for all $b \in B$, and by tightness of M, we must have $w_a \in [M, w]_b$ for all $b \in B$. This means that $w_a \in [M, w]_B$, and thus, that $M, w \nvDash D_B\varphi$, as desired. □

1.2.2 Tightness and Saturation

Tightness is familiar from modal logic (see, for example, [14]). Epistemic saturation seems new, but turns out to be equivalent with another very familiar notion from modal logic, namely, modal saturation (see, for example, [7]). A model $M = (W, R, V)$ is modally saturated iff for every $w \in W$, every $a \in \mathcal{A}$, and every set of $\mathcal{L}_K$-formulas Σ, if every finite subset of Σ is satisfiable in $[M, w]_a$, then Σ itself is satisfiable in $[M, w]_a$.

Proposition 1.4 *A model is epistemically saturated iff it is modally saturated.*

Proof. Let $M = (W, R, V)$ be a model, $a \in \mathcal{A}$, and $w \in W$.

($\Rightarrow$) Suppose M is epistemically saturated. Let Σ be a set of $\mathcal{L}_K$-formulas such that every finite subset of Σ is satisfiable in $[M, w]_a$. Then every finite subset of $\Sigma' = \Sigma \cup \mathrm{KS}_a(M, w)$ is also satisfiable in $[M, w]_a$. It follows, by compactness, that Σ' is satisfiable and therefore consistent. Then, by epistemic saturation of M, Σ' must be satisfiable in $[M, w]_a$. This means that Σ is satisfiable in $[M, w]_a$, and thus that M is modally saturated.

($\Leftarrow$) Suppose M is modally saturated. Let Σ be a set of $\mathcal{L}_K$-formulas that is consistent with $\mathrm{KS}_a(M, w)$. Then every finite subset of Σ is satisfiable in $[M, w]_a$. To see this, let Σ' be a finite subset of Σ and suppose that Σ' is *not* satisfiable in $[M, w]_a$. Then $\neg \bigwedge \Sigma' \in \mathrm{KS}_a(M, w)$, which contradicts the assumption that Σ is consistent with $\mathrm{KS}_a(M, w)$. So every finite subset of Σ is satisfiable in $[M, w]_a$. By modal saturation, it follows that Σ is satisfiable in $[M, w]_a$, and thus that M is epistemically saturated. □

Henceforth, we will simply refer to modal and epistemic saturation as *saturation.*

1.2.3 Tight & Saturated versus Finite & Distinguishing

Next, we show how tightness and saturation are related to the special model properties defined by van der Hoek, van Linder, and Meyer. We prove that every finite, distinguishing model is tight and saturated, and also identify an interesting intermediate class of models.

Proposition 1.5 *Finite, distinguishing models are saturated and tight.*

Proof. Finite models are always saturated [7]. Now suppose a model M is finite and therefore saturated, but *not* tight. Then for some agent a there must be two worlds w and v in M such that everything a knows in w is true in v, but v is not contained in $[M, w]_a$. Let Γ be the set of formulas true in v. Clearly, Γ is consistent with $\mathrm{KS}_a(M, w)$. M is saturated so there must be a world u in $[M, w]_a$ that satisfies all formulas in Γ. But this means that u and v satisfy exactly the same formulas. So M is not distinguishing. We conclude that, if M is finite and distinguishing, then it must be saturated and tight. □

The class of saturated and distinguishing models clearly subsumes the class of finite and distinguishing models, but is itself subsumed by the class of tight and saturated models. Interestingly, a model is saturated and distinguishing if and only if it is a generated submodel of the so-called *canonical* model. To show this, let us first recall the relevant definitions, which are all standard in modal logic [7].

A set of $\mathcal{L}_K$-formulas Σ is *maximally consistent* iff it is consistent and for all $\varphi \in \mathcal{L}_K$, either $\varphi \in \Sigma$ or $\neg\varphi \in \Sigma$. Note that the theory of a pointed model is always maximally consistent. The *canonical model* M^c is a triple (W^c, R^c, V^c) where:

$$\begin{array}{lcl} W^c & = & \{w_\Sigma \mid \Sigma \text{ is a maximally consistent set of } \mathcal{L}_K\text{-formulas}\} \\ R^c(a) & = & \{(w_\Sigma, w_\Delta) \mid K_a\varphi \in \Sigma \text{ implies } \varphi \in \Delta\} \\ V^c(w_\Gamma) & = & \{p \in \mathcal{P} \mid p \in \Sigma\} \end{array}$$

It is a well-known result that for every $\varphi \in \mathcal{L}_K$, $M^c, w_\Sigma \models \varphi$ iff $\varphi \in \Sigma$.
Given a model $M = (W, R, V)$ and a set of worlds $X \subseteq W$, the model $M|X = (X, R|X, V|X)$ is called the *restriction* of M to X. $M|X$ is a *generated submodel* of M iff, whenever $w \in X$ and $v \in [M, w]_a$ for some $a \in \mathcal{A}$, then also $v \in X$. So a generated submodel is a submodel that preserves information states.
Finally, two models (M, w) and (M', w') are isomorphic, $M, w \cong M', w'$, iff there is a *bijective* bisimulation (i.e., an *isomorphism*) $\mathcal{Z}$ between M and M' such that $w\mathcal{Z}w'$.

Proposition 1.6 *A model is saturated and distinguishing iff it is isomorphic to a generated submodel of the canonical model.*

Proof. Let $M = (W, R, V)$ be a model.

($\Rightarrow$) Suppose M is saturated and distinguishing. Then, every world in M has a unique, maximally consistent theory, and therefore uniquely corresponds to a world in M^c. Let W' be the set of all worlds w' in M^c that correspond to a world w in M, and let $M' = (W', R', V')$ be the restriction of M^c to W'. We show (1) that M' is a generated submodel of M^c and (2) that M and M' are isomorphic.

For (1), suppose that $w' \in W'$ and $(w', v') \in R^c(a)$ for some $a \in \mathcal{A}$ and some $v' \in W^c$. We must show that $v' \in W'$. By definition of the canonical model v' satisfies every formula in $\mathrm{KS}_a(M^c, w')$. So $\Gamma(M^c, v')$ is consistent with $\mathrm{KS}_a(M^c, w')$, and therefore also with $\mathrm{KS}_a(M, w)$. Then, by epistemic saturation of M, $\Gamma(M^c, v')$ must be satisfiable in $[M, w]_a$. But this means that $[M, w]_a$ must contain a world v with $\Gamma(M^c, v')$ as its theory. It follows that v' is in W', and thus, that M' is a generated submodel of $M_{\mathcal{A},\mathcal{P}}$.

For (2), let $\mathcal{Z}$ be the bijection that relates every world $w \in W$ with its corresponding world $w' \in W'$. Clearly, w and w' satisfy the same proposition letters. It remains to check the back and forth clauses. We do the forth clause; the back clause is completely analogous. Let $a \in \mathcal{A}$ and $v \in [M, w]_a$. Clearly, $\Gamma(M, v)$ is consistent with $\mathrm{KS}_a(M, w)$, and therefore $\Gamma(M', v')$ is also consistent with $\mathrm{KS}_a(M, w)$. But then, by definition of the canonical model, $v' \in [M', w']_a$, as desired.

($\Leftarrow$) Suppose M is isomorphic to a generated submodel $M' = (W', R', V')$ of M^c. It is clear that no two worlds in M can have the same theory, so M must be distinguishing. To show that M is saturated it suffices to show that M' is saturated. M^c is well-known to be saturated [7]. Moreover, generated submodels it preserve information states and saturation thereof. So M' and M are saturated. □

We are not aware of any earlier statements of this result, although Fine [14] made several observations in this direction, and we would not be surprised if others had proven similar results, be it for different purposes and presumably in terms of modal saturation rather than epistemic saturation.

1.2.4 A Complete Characterization of Full Communication

So far, we have established that finite, distinguishing models are canonical, that canonical models are saturated and tight, and that saturation and tightness are sufficient conditions for compliance with the principle of full communication. Now we could ask whether they are also *necessary*, that is, whether *all* models in which distributed knowledge complies with the principle of full communication are saturated and tight. The answer is *no.* To see this, reconsider the proof of proposition 1.3. The role of saturation here is to ensure that every $\mathcal{L}_K$-formula φ that is consistent with the knowledge set of a group of agents is satisfiable in the information state of every agent in the group. Saturation is sufficient here, but not necessary: it is concerned with *sets* of formulas, where it could equally well just be concerned with *single* formulas. Then, once it has been established that φ is satisfiable in the information state of every agent in the group, the role of tightness is to ensure that there is at least one world satisfying φ which is contained in the information state of *all* agents in the group. Again, tightness is sufficient to fulfil this role, but not necessary. These observations give rise to the following definition:

Definition 1.7 (Full Communication Model) *A model $M = (W, R, V)$ is a full communication model iff for all $w \in W$ and all $B \subseteq \mathcal{A}$, every $\mathcal{L}_K$-formula that is consistent with $KS_B(M, w)$ is satisfiable in $[M, w]_B$.*

Notice that full models, as defined by Gerbrandy, are full communication models: where Gerbrandy quantifies over *sets of $\mathcal{L}_D$-formulas*, we quantify over *single $\mathcal{L}_K$-formulas.* The following proposition establishes that tight and saturated models are full communication models as well.

Proposition 1.8 *Tight and saturated models are full communication models.*

Proof. Let M be tight and saturated, w in M, $B \in \mathcal{A}$, and Σ a set of $\mathcal{L}_K$-formulas that is consistent with $\mathrm{KS}_B(M, w)$. We must show that Σ is satisfiable in $[M, w]_B$. Let $a \in B$. Then $\Sigma \cup \mathrm{KS}_B(M, w)$ is consistent with $\mathrm{KS}_a(M, w)$, so by saturation of M, $[M, w]_a$ must contain a world w_a that satisfies all formulas in $\Sigma \cup \mathrm{KS}_B(M, w)$. Thus, w_a is consistent with $\mathrm{KS}_b(M, w)$ for all $b \in B$, and by tightness of M, we must have $w_a \in [M, w]_b$ for all $b \in B$. This means that $w_a \in [M, w]_B$. So Σ is satisfiable in $[M, w]_B$ which means that M is a full communication model. □

Now we prove that distributed knowledge complies with the principle of full communication in full communication models, and moreover, that full communication models are the *only* models in which distributed knowledge complies with the principle of full communication.

Proposition 1.9 *Distributed knowledge complies with the principle of full communication in M if and only if M is a full communication model.*

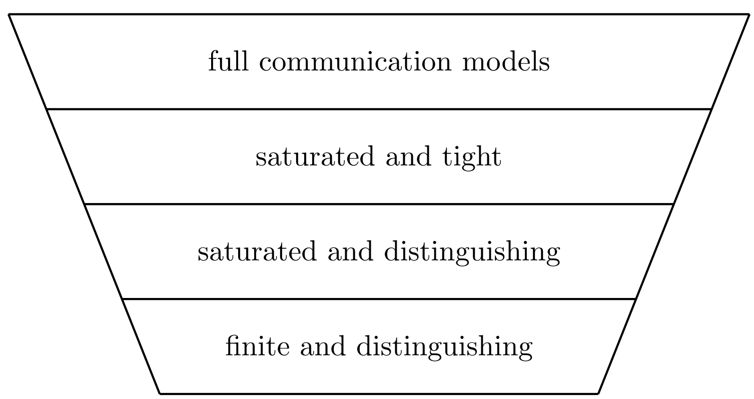
Proof. The *if* part was, implicitly, already established by Gerbrandy [16]: suppose M is a full communication model, w a world in M, B a group of agents, and φ a formula in $\mathcal{L}_K$ that is *not* entailed by $\mathrm{KS}_B(M, w)$. Then $\neg\varphi$ is consistent with $\mathrm{KS}_B(M, w)$, and therefore, as M is a full communication model, $\neg\varphi$ is satisfiable in $[M, w]_B$. But this means that φ cannot be distributed knowledge among B in (M, w). So distributed knowledge complies with the principle of full communication in M.

For the *only if* part, suppose that M is *not* a full communication model. Then, for some w in M and some $B \subseteq \mathcal{A}$, there is a formula φ in $\mathcal{L}_K$ that is consistent with $\mathrm{KS}_B(M, w)$ but not satisfiable in $[M, w]_B$. For this to be the case, $\neg\varphi$ must not be entailed by $\mathrm{KS}_B(M, w)$. On the other hand, $\neg\varphi$ must hold throughout $[M, w]_B$ and therefore be distributed knowledge among B in (M, w). So distributed knowledge does *not* comply with the principle of full communication in M. □

1.2.5 Conclusion

Van der Hoek, van Linder, and Meyer [43] as well as Gerbrandy [16] observed that distributed knowledge does not always comply with the principle of full communication. They also observed that distributed knowledge *does* comply with the principle in finite, distinguishing models, in locally distinguishing models, and in full models. It was not clear, however, why these properties were sufficient and whether they were necessary for compliance with the principle of full communication.

We established the following hierarchy of models in which distributed knowledge complies with the principle of full communication:



Our point of departure was the class of saturated and tight models, in which every agent's information state contains all worlds that are consistent with her knowledge set. We established that this class contains all finite and distinguishing models and all canonical (saturated and distinguishing) models. On the other hand, it is subsumed by the class of full communication models, which was shown to consist of all models in which distributed knowledge complies with the principle of full communication. Interestingly, the characteristic property of full communication models turned out to be a weak version of Gerbrandy's fullness property. One more thing to notice about the hierarchy above is that less restrictive properties are concerned with more fine-grained structural aspects of a model. Finiteness and distinguishability are concerned with the structure of a model as a whole, saturation and tightness are concerned with information states of individual agents, and full communication models impose restrictions on the information states of groups of agents. Distributed knowledge itself is defined in terms of information states of groups of agents. In the light of this observation it is not so surprising that finiteness, distinguishability, tightness, and saturation only led to a partial characterization of the class of full communication models.

As a final remark we would like to point out that the principle of full communication considered here presupposes a particular kind of communication, namely *private* communication. Other kinds of communication may be considered as well. Van Benthem [39], for example, is interested in the information that can be obtained by a group of agents through *public* communication. Parikh and Pacuit [25] are interested in the information that can be established by a group of agents relative to a restricted communication network. Chapter 4 and 5 of [28] are also relevant in this respect. In general, a dynamic view on epistemic logic [46] gives rise to many interesting notions of group knowledge, most of which remain to be explored. We leave such explorations for another occasion, however, and now turn to the second topic of the chapter.

1.3 Expressive Power

Bisimulation, which was introduced in a general modal setting by van Benthem [37], is a standard measure of structural equivalence between epistemic models. Intuitively, two pointed models are bisimilar iff (1) they assign the same truth values to all proposition letters and (2) they assign equivalent information states to all agents.

Definition 1.10 (Bisimulation) *Let* $M = (W, R, V)$ *and* $M' = (W', R', V')$

be two epistemic models. A non-empty relation $\mathcal{Z} \subseteq W \times W'$ is a bisimulation between M and M' iff for every $w \in W$ and $w' \in W'$ such that $w\mathcal{Z}w'$ we have:

1. $V(w) = V'(w')$
2. *For every agent $a \in \mathcal{A}$:*

$$\begin{array}{llll} \textbf{forth} & \forall v \in [M,w]_a & : \exists v' \in [M',w']_a & : v\mathcal{Z}v' \\ \textbf{back} & \forall v' \in [M',w']_a & : \exists v \in [M,w]_a & : v\mathcal{Z}v' \end{array}$$

Two pointed models (M,w) and (M',w') are bisimilar, $(M,w) \simeq (M',w')$, if and only if there is a bisimulation $\mathcal{Z}$ between M and M' such that $w\mathcal{Z}w'$.

Bisimilarity matches the expressive power of the basic epistemic language: if $(M,w) \simeq (M',w')$ then (M,w) and (M',w') satisfy the same $\mathcal{L}_K$-formulas. The language with distributed knowledge operators, however, is more expressive. To see this, consider the models depicted in figure 1.1 and 1.2 (where all accessibility relations are equivalence relations).

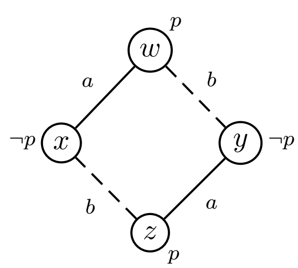


Figure 1.1: Model M.

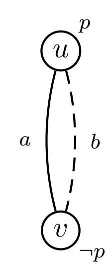


Figure 1.2: Model N.

(M,w) and (N,u) are bisimilar, but do not satisfy the same formulas in $\mathcal{L}_D$. For example, with $B = \{a,b\}$ we have:

$$\begin{array}{lcl} M,w & \models & D_B p \\ N,u & \not\models & D_B p \end{array}$$

Is there a natural extended notion of bisimulation that matches the expressive power of distributed knowledge operators? This question is the first of a list of open problems recently put together by van Benthem [40]. Section 1.3.1 proposes an extended notion of bisimulation, and proves its adequacy. Section 1.3.2 investigates related model comparison games, and section 1.3.3 explores a corresponding notion of saturation.

1.3.1 Collective Bisimulation

Bisimilarity requires two models to associate equivalent information states with every single agent. But distributed knowledge involves information states of *groups* of agents. Thus, in order to obtain a suitable notion of bisimulation, it seems natural to require two models to associate equivalent information states not just with every individual agent, but with every *group* of agents.

Definition 1.11 (Collective Bisimulation) *Let $M = (W, R, V)$ and $M' = (W', R', V')$ be two models. A non-empty relation $\mathcal{Z} \subseteq W \times W'$ is a collective bisimulation between M and M' iff for every $w \in W$ and $w' \in W'$ such that $w\mathcal{Z}w'$:*

1. $V(w) = V'(w')$
2. *For every* group *of agents $B \subseteq \mathcal{A}$:*

$$\begin{array}{llll} \textbf{forth} & \forall v \in [M,w]_B & : \exists v' \in [M',w']_B & : v\mathcal{Z}v' \\ \textbf{back} & \forall v' \in [M',w']_B & : \exists v \in [M,w]_B & : v\mathcal{Z}v' \end{array}$$

Two models (M, w) and (M', w') are collectively bisimilar, $(M, w) \simeq_c (M', w')$, iff there is a collective bisimulation $\mathcal{Z}$ between M and M' such that $w\mathcal{Z}w'$.

Collective bisimilarity generalizes ordinary bisimilarity. The latter only requires the back and forth conditions to hold for singleton groups of agents. So if two models are collectively bisimilar, then they are also bisimilar, but not vice versa: the models depicted in figure 1.1 and 1.2, for example, are bisimilar, but not collectively bisimilar.

We now show that collective bisimulation indeed matches the expressive power of $\mathcal{L}_D$ and establish some of its basic properties.

Proposition 1.12 (Adequacy) *If two pointed models are collectively bisimilar, then they satisfy exactly the same formulas in $\mathcal{L}_D$.*

Proof. Let $M = (W, R, V)$ and $M' = (W', R', V')$ be two models and let $w \in W$ and $w' \in W'$ be such that $(M, w) \simeq_c (M', w')$. We must prove that for every $\varphi \in \mathcal{L}_D$:

$$M, w \models \varphi \quad \text{iff} \quad M', w' \models \varphi$$

Clearly, it suffices to prove either the *if* or the *only if* part of the statement. We do the latter by induction on the complexity of φ. The base case and the induction steps for negation, conjunction and individual knowledge operators are standard [7]. Here is the induction step for distributed knowledge operators. Let φ be of the form $D_B\psi$ and suppose that $M, w \not\models D_B\psi$. Then $M, v \models \neg\psi$ for some $v \in [M, w]_B$. As $(M, w) \simeq_c (M', w')$, there is a world $v' \in [M', w']_B$ such that $(M, v) \simeq_c (M', v')$. By the induction hypothesis, then, $M', v' \models \neg\psi$, and thus $M', w' \not\models D_B\psi$. □

Proposition 1.13 *Collective bisimulation is an equivalence relation.*

Proof. Reflexivity and symmetry are clear. Transitivity amounts to:

$$(M, w) \simeq_c (M', w') \text{ and } (M', w') \simeq_c (M'', w'') \text{ imply } (M, w) \simeq_c (M'', w'')$$

Let $\mathcal{Z}$ be a collective bisimulation between (M, w) and (M', v'), and let $\mathcal{Z}'$ be a collective bisimulation between (M', w') and (M'', v''). Then the relation $\mathcal{Z}''$ between worlds in M and worlds in M'' given by $\{(w, w'') \mid \exists w' \in M' : w\mathcal{Z}w' \text{ and } w'\mathcal{Z}'w''\}$ is a collective bisimulation between (M, w) and (M'', v''). To see this, first observe that (M, w) and (M'', v'') should satisfy the same proposition letters, by $(M, w) \simeq_c (M', w')$ and $(M', w') \simeq_c (M'', w'')$. Now let v be a world in $[M, w]_B$ for some arbitrary group of agents B. Then, by $(M, w) \simeq_c (M', w')$, there is a world v' in $[M', w']_B$ such that $v\mathcal{Z}v'$. Moreover, by $(M', w') \simeq_c (M'', w'')$, there is a world v'' in $[M'', w'']_B$ such that $v'\mathcal{Z}'v''$. Then, by definition of $\mathcal{Z}''$, we have $v\mathcal{Z}''v''$. This establishes the forth clause. The back clause can be proven analogously. So we may conclude that $\mathcal{Z}''$ is a collective bisimulation between (M, w) and (M'', w''). □

Proposition 1.14 *The set of collective bisimulations between any two models is closed under taking arbitrary (finite or infinite) unions.*

Proof. Let $\{\mathcal{Z}_i \mid i \in I\}$ be a non-empty set of collective bisimulations between two models M and M'. To show that $\mathcal{Z} = \bigcup_{i \in I} \mathcal{Z}_i$ is again a collective bisimulation between M and M', let w in M and w' in M' be any two worlds such that $w\mathcal{Z}w'$. Then we must have $w\mathcal{Z}_iw'$ for some $i \in I$. Now let v be a world in $[M, w]_B$ for some arbitrary group of agents B. Then, there must be a world v' in $[M', w']_B$ such that $v\mathcal{Z}_iv'$. But $v\mathcal{Z}_iv'$ means that $v\mathcal{Z}v'$ holds as well. This establishes the forth clause, and the back clause can be proven analogously. We conclude that $\mathcal{Z}$ is a collective bisimulation between M and M'. □

Corollary 1.15 *If there is a collective bisimulation between two models M and M', then there is always a* maximal *collective bisimulation between M and M': one that includes all other collective bisimulations between M and M'.*

Proof. Take the union of all collective bisimulations between M and M'. By proposition 1.14, this is again a collective bisimulation between M and M' and clearly, it includes all others. □

1.3.2 Model Comparison Games

If two models are bisimilar, then they satisfy exactly the same formulas in $\mathcal{L}_K$. The opposite is only true for certain special classes of models. If two *finite* models, for example, satisfy the same formulas in $\mathcal{L}_K$, then they are bisimilar. However, *infinite* models may very well satisfy exactly the same formulas in $\mathcal{L}_K$ even though they are not bisimilar. Intuitively, this is because whether or not a model (M, w) satisfies a formula φ only depends on worlds in M that can be reached from w in a finite number of steps along the accessibility relations in M. This number of relevant steps is bounded by the so-called *modal depth* of φ.

Definition 1.16 (Modal Depth) *The modal depth $d(\varphi)$ of a formula $\varphi \in \mathcal{L}_K$ is defined recursively as follows:*

$$\begin{array}{lcll} d(p) & = & 0 & \textit{for all } p \in \mathcal{P} \\ d(\neg\varphi) & = & d(\varphi) & \\ d(\varphi \wedge \psi) & = & max(d(\varphi), d(\psi)) & \\ d(K_a\varphi) & = & d(\varphi) + 1 & \textit{for all } a \in \mathcal{A} \end{array}$$

Now, for two models (M, w) and (M', w') to satisfy the same formulas in $\mathcal{L}_K$ it is sufficient that (M, w) and (M', w') satisfy the same proposition letters and that for every *finite* path starting from w in M, we can find a corresponding path starting from w' in M' (and vice versa). Bisimulation requires something stronger, namely that (M, w) and (M', w') satisfy the same proposition letters and that for every (possibly *infinite*) path starting from w in M, we can find a corresponding path starting from w' in M' (and vice versa). This is why two infinite models that satisfy the same formulas in $\mathcal{L}_K$ are not necessarily bisimilar.

Model comparison games can be seen as finite approximations of a bisimulation. A model comparison game is played on two models, say (M, w) and

(M', w'), by two players called spoiler and duplicator, and consists of a fixed number of rounds. Spoiler tries to establish that (M, w) and (M', w') satisfy different formulas; duplicator tries to show that they satisfy exactly the same formulas. The number of rounds of the game yields a bound on the length of the paths starting from w in M and from w' in M' that are available to spoiler as possible evidence for a structural difference between M and M'. If spoiler cannot win the n-round model comparison game on (M, w) and (M', w'), then these models satisfy exactly the same formulas in $\mathcal{L}_K$ up to depth n. So if duplicator has a winning strategy for all model comparison games on (M, w) and (M', w'), then (M, w) and (M', w') satisfy the same $\mathcal{L}_K$-formulas of arbitrary depth. And now the converse also holds: if (M, w) and (M', w') satisfy the same $\mathcal{L}_K$-formulas, then duplicator has a winning strategy for all model comparison games on (M, w) and (M', w'). This works out because games themselves have a finite number of rounds (just like formulas have finite modal depth), but for every n, there is a game with n rounds (just as for every n, there are formulas with modal depth n). Model comparison games characteristic for $\mathcal{L}_K$ are standard and can be found, for example, in [46]. Here, we define model comparison games characteristic for $\mathcal{L}_D$, called $\mathcal{L}_D$-games, and show that duplicator has a winning strategy in all $\mathcal{L}_D$-games on two models iff those two models satisfy exactly the same $\mathcal{L}_D$-formulas. Just as $\mathcal{L}_K$-games can be seen as finite approximations of a bisimulation, $\mathcal{L}_D$-games can be seen as finite approximations of a collective bisimulation.

Definition 1.17 ($\mathcal{L}_D$-games) *Let $M = (W, R, V)$ and $M' = (W', R', V')$ be two models, let $w \in W$ and let $w' \in W'$. The rules of the n-round $\mathcal{L}_D$-game on (M, w) and (M', w') are as follows:*

1. *If $n = 0$ then duplicator wins iff $V(w) = V'(w')$.*
2. *If n**not** $= 0$ then spoiler can do either one of the following moves:*

   forth-move *Spoiler picks a group of agents $B \subseteq \mathcal{A}$ and a world $v \in [M, w]_B$. Duplicator responds by picking a world $v' \in [M', w']_B$. The rest of the game is the $(n-1)$-round $\mathcal{L}_D$-game on (M, v) and (M', v').*

   back-move *Spoiler picks a group of agents $B \subseteq \mathcal{A}$ and a world $v' \in [M', w']_B$. Duplicator responds by picking a world $v \in [M, w]_B$. The rest of the game is the $(n-1)$-round $\mathcal{L}_D$-game on (M, v) and (M', v').*

3. If a player cannot make any further move, she loses the game.

Proposition 1.18 *Let (M, w) and (M', w') be two models, and let $\mathcal{P}$ be finite. Then (M, w) and (M', w') satisfy exactly the same $\mathcal{L}_D$-formulas iff for all $n \geq 0$, duplicator has a winning strategy for the n-round $\mathcal{L}_D$-game on (M, w) and (M', w').*

Proof. Define the depth $d(\varphi)$ of a formula $\varphi \in \mathcal{L}_D$ just as for formulas in $\mathcal{L}_K$ (see definition 1.16) with the following additional clause:

$$d(D_B\varphi) = d(\varphi) + 1 \text{ for all } B \subseteq \mathcal{A}$$

Write $(M, w) \equiv_n (M', w')$ iff (M, w) and (M', w') satisfy exactly the same formulas in $\mathcal{L}_D$ with depth n. Observe that, as $\mathcal{P}$ and $\mathcal{A}$ are both assumed to be finite, for every $n \geq 0$, there are only finitely many formulas with depth n, up to logical equivalence. Now we prove, by induction on n, that duplicator has a winning strategy for the n-round $\mathcal{L}_D$-game on (M, w) and (M', w') iff $(M, w) \equiv_n (M', w')$.

The base case $(n = 0)$ follows directly from the definitions.

The induction step involves two directions, which we treat separately:

($\Rightarrow$) Suppose duplicator has a winning strategy for the $(n+1)$-round $\mathcal{L}_D$-game on (M, w) and (M', w'). Then we must show that for all $\varphi \in \mathcal{L}_D$ such that $d(\varphi) = n + 1$, we have $M, w \vDash \varphi$ iff $M', w' \vDash \varphi$. We do so by induction on φ, and only treat the non-standard case in which φ is of the form $D_B\phi$, where $d(\phi) = n$. Suppose $M, w \vDash D_B\phi$. We show that $M', w' \vDash D_B\phi$ must hold as well: let v' be any world in $[M', w']_B$. Suppose spoiler chooses v' in a first back-move of the $(n + 1)$-round $\mathcal{L}_D$-game on (M, w) and (M', w'). Then, by assumption, duplicator can pick a world v in $[M, w]_B$ such that she (duplicator) has a winning strategy for the remaining n-round $\mathcal{L}_D$-game on (M, v) and (M', v'). By the induction hypothesis, $(M, v) \equiv_n (M', v')$. In particular, we have that $M', v' \vDash \phi$, and as v' was an arbitrary world in $[M, w]_B$, we may conclude that $M', w' \vDash D_B\phi$.

($\Leftarrow$) Now suppose that $(M, w) \equiv_{n+1} (M', w')$. We must show that duplicator has a winning strategy in the $(n + 1)$-round $\mathcal{L}_D$-game on (M, w) and (M', w'). Suppose spoiler starts with a back-move and picks a world v' in $[M', w']_B$. Now, toward a contradiction, suppose that there is no world v in $[M, w]_B$ such that $(M, v) \equiv_n (M', v')$ (which is what duplicator needs for a winning strategy). Then, for every v in $[M, w]_B$, there is a formula $\psi_v \in \mathcal{L}_D$ of depth n such that $M, v \vDash \psi_v$ but $M', v' \nvDash \psi_v$. Define:

$$\psi_w = \bigvee_{v \in [M,w]_B} \psi_v$$

We observed that there are only finitely many formulas with depth n, up to logical equivalence. So ψ_w corresponds to a finite disjunction, again of depth n. We have $M, w \models D_B\psi_w$, but $M', w' \not\models D_B\psi_w$, even though $d(D_B\psi_w) = n+1$. This contradicts our assumption that $(M, w) \equiv_{n+1} (M', w')$. So we may conclude that duplicator has a winning strategy for the $(n+1)$-round $\mathcal{L}_D$-game on (M, w) and (M', w'). □

1.3.3 Saturation and Fullness

Recall the notions of saturation and fullness from section 1.2. A model is saturated iff every set of $\mathcal{L}_K$-formulas that is consistent with the knowledge set of a single agent is satisfiable in the information state of that single agent. A model is full iff every set of $\mathcal{L}_D$-formulas that is consistent with the knowledge set of a group of agents is satisfiable in the information state of that group of agents. Notice that fullness generalizes saturation just like collective bisimulation generalizes ordinary bisimulation and distributed knowledge operators generalize individual knowledge operators: where saturation, bisimulation, and individual knowledge operators are concerned with the information states of individual agents, fullness, collective bisimulation and distributed knowledge operators are concerned with the information states of *groups* of agents.

There is an important and well-known connection between saturation, bisimilarity, and the expressive power of $\mathcal{L}_K$: saturated models are bisimilar iff they satisfy the same $\mathcal{L}_K$-formulas [7]. The following proposition establishes a similar connection between fullness, collective bisimilarity, and the expressive power of $\mathcal{L}_D$.

Proposition 1.19 *Full models are collectively bisimilar iff they satisfy the same $\mathcal{L}_D$-formulas.*

Proof. Proposition 1.12 already established that, in general, collectively bisimilar models satisfy the same formulas in $\mathcal{L}_D$. Here, we show that full models which satisfy the same formulas in $\mathcal{L}_D$ are collectively bisimilar. Let M and M' be full, and let w in M and w' in M' be such that (M, w) and (M', w') satisfy the same formulas in $\mathcal{L}_D$. Let $\mathcal{Z}$ be the relation between W and W' that consists of all pairs of worlds that satisfy exactly the same formulas in $\mathcal{L}_D$. Clearly, $w\mathcal{Z}w'$. We will show that $\mathcal{Z}$ is a collective bisimulation between M and M'. To do so, let B be an arbitrary group of agents and let v be any world in $[M, w]_B$. Let Γ be the set of $\mathcal{L}_D$-formulas true in

v. Then Γ is consistent with $\mathrm{KS}_B(M,w)$, and therefore also consistent with $\mathrm{KS}_B(M',w')$. As M' is full, Γ must be satisfiable in $[M',w']_B$. But this means that there must be a world v' in $[M',w']_B$ that satisfies exactly the same $\mathcal{L}_D$-formulas as v. So we have $v\mathcal{Z}v'$, which means that $\mathcal{Z}$ is indeed a collective bisimulation between M and M'. □

This can be taken a bit further. The following propositions establish that full models are collectively bisimilar iff they are bisimilar and that with respect to full models, $\mathcal{L}_D$ has no more expressive power than $\mathcal{L}_K$.

Proposition 1.20 *Full models are bisimilar iff they are collectively bisimilar.*

Proof. Clearly, collectively bisimilar models are always bisimilar. We must show that full, bisimilar models are always collectively bisimilar. To do so, let M and M' be full and let w in M and w' in M' be such that (M,w) and (M',w') are bisimilar. Let $\mathcal{Z}$ be the *maximal* bisimulation between (M,w) and (M',w') (the existence of which is established in [7]). We will show that $\mathcal{Z}$ is also a *collective* bisimulation between (M,w) and (M',w'). To do so, let B be an arbitrary group of agents and let v be any world in $[M,w]_B$. Let Γ be the set of $\mathcal{L}_D$-formulas true in v. Then Γ is consistent with $\mathrm{KS}_B(M,w)$. (M,w) and (M',w') are bisimilar, so they satisfy the same formulas in $\mathcal{L}_K$. Therefore, Γ is also consistent with $\mathrm{KS}_B(M',w')$. M' is full, so Γ must be satisfiable in $[M',w']_B$. This means that there must be a world v' in $[M',w']_B$ that satisfies exactly the same $\mathcal{L}_D$-formulas as v. In particular, v and v' satisfy exactly the same $\mathcal{L}_K$-formulas. M and M' are both saturated, so (M,v) and (M',v') must be bisimilar. But then, as $\mathcal{Z}$ was assumed to be maximal, we must have $v\mathcal{Z}v'$, and this establishes that $\mathcal{Z}$ is indeed a collective bisimulation. □

Proposition 1.21 *Full models satisfy the same formulas in $\mathcal{L}_K$ iff they satisfy the same formulas in $\mathcal{L}_D$.*

Proof. Clearly, it is enough to prove the *only if* part. Let (M,w) and (M',w') be full. Suppose (M,w) and (M',w') satisfy the same formulas in $\mathcal{L}_K$. Since both models are saturated, they must be bisimilar [7]. Then, by proposition 1.20, they must be collectively bisimilar. But then it follows from proposition 1.12 that they must satisfy exactly the same formulas in $\mathcal{L}_D$. □

1.3.4 Conclusion

We have proposed an extended notion of bisimulation that matches the expressive power of the basic epistemic language extended with distributed knowledge operators. We have also defined related model comparison games and established their adequacy. Finally, we showed that fullness generalizes saturation just as collective bisimulation generalizes ordinary bisimulation and established that full models are collectively bisimilar iff they are bisimilar iff they satisfy exactly the same $\mathcal{L}_K$-formulas iff they satisfy exactly the same $\mathcal{L}_D$-formulas. In particular, $\mathcal{L}_D$ and $\mathcal{L}_K$ have equal expressive power with respect to full models.

As mentioned before, the issue addressed here is the first of a list of open problems in a recent survey by van Benthem [40]. The model comparison games defined here are variations on standard games for the basic modal language as described, for instance, by van Ditmarsch, van der Hoek, and Kooi in [46]. The connection between saturation, bisimulation, and the expressive power of $\mathcal{L}_K$ is a standard result in modal logic [7].

Chapter 2

Distributed Knowledge in Dynamic Epistemic Logic

Dynamic epistemic logic deals with information *change*. It is the subject of a lively field of research, with key contributions by Gerbrandy [16], Baltag, Moss, and Solecki [3, 4], and van Ditmarsch [44]. A particularly attractive system, called the *logic of communication and change*, LCC, has recently been proposed by van Benthem, van Eijck, and Kooi in [42]. It takes propositional dynamic logic as its static point of departure, but does not consider intersection modalities, which are needed to define distributed knowledge. In this chapter, we show that intersection modalities can be incorporated into the logic of communication and change in a straightforward way. We will define LCC as in [42], merely highlighting the slight modifications involved in adding intersection modalities. This chapter is meant to be read along with [42]. The underlying intuitions provided there are not repeated here.

2.1 Propositional Dynamic Logic

Language The basic language of propositional dynamic logic, PDL, consists of formulas φ and relational symbols π:

$$\begin{array}{rcl} \varphi & := & \top \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid [\pi]\varphi \\ \pi & := & a \mid ?\varphi \mid \pi_1;\pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^* \end{array}$$

This language can be extended with so-called intersection modalities. That is, if π_1 and π_2 are relational symbols, then $\pi_1 \cap \pi_2$ is also a relational symbol. The resulting language is denoted as $\mathsf{PDL}^{\cap}$.

Semantics. The semantics of formulas in PDL and PDL$^\cap$ is given in terms of the models defined in the previous chapter, which are hereafter called *epistemic models*. Formulas are interpreted as subsets of W, and relational symbols are interpreted as binary relations on W:

$$
\begin{array}{lcl}
[[\top]]^M & = & W \\
[[p]]^M & = & \{w \in W \mid p \in V(w)\} \\
[[\neg\varphi]]^M & = & W \setminus [[\varphi]]^M \\
[[\varphi_1 \wedge \varphi_2]]^M & = & [[\varphi_1]]^M \cap [[\varphi_2]]^M \\
[[[\pi]\varphi]]^M & = & \{w \in W \mid (w,v) \in [[\pi]]^M \text{ implies } v \in [[\varphi]]^M\} \\
 & & \\
[[a]]^M & = & R(a) \\
[[?\varphi]]^M & = & \{(w,w) \in W \times W \mid w \in [[\varphi]]^M\} \\
[[\pi_1;\pi_2]]^M & = & [[\pi_1]]^M \circ [[\pi_2]]^M \\
[[\pi_1 \cup \pi_2]]^M & = & [[\pi_1]]^M \cup [[\pi_2]]^M \\
[[\pi_1 \cap \pi_2]]^M & = & [[\pi_1]]^M \cap [[\pi_2]]^M \\
[[\pi^*]]^M & = & ([[\pi]]^M)^*
\end{array}
$$

$[[\pi_1]]^M \circ [[\pi_2]]^M$ is the composition of $[[\pi_1]]^M$ and $[[\pi_2]]^M$, and $([[\pi]]^M)^*$ is the reflexive transitive closure of $[[\pi]]^M$. PDL$^\cap$ is decidable [10] and has a complete axiomatization [2].

It is possible to describe various kinds of group knowledge in PDL$^\cap$. $[\bigcup_{a\in B} a]\varphi$, for example, means that φ holds in all worlds that any agent in B considers possible, i.e., that every agent in B knows φ. This is traditionally written as $E_B\varphi$. $[(\bigcup_{a\in B} a)^*]\varphi$ means that every agent in B knows φ, that every agent in B knows that every agent in B knows φ, and so on, i.e., that φ is *common knowledge* among B. This is traditionally written as $C_B\varphi$. Finally, $[\bigcap_{a\in B} a]\varphi$ means that φ holds in every world that *all* agents in B consider possible, i.e., that φ is *distributed knowledge* among B. This, as we saw in the previous chapter, is traditionally written as $D_B\varphi$.

2.2 Update Models

In LCC, *epistemic models* are used to represent the agents' information, and so-called *update models* are used to represent information bearing *events*. To define update models we need the following notion:

Substitutions. A substitution for a language $\mathcal{L}$ is a function of type $\mathcal{L} \to \mathcal{L}$ that distributes over all language constructs, and maps all but a finite number of proposition letters to themselves. Let ϵ denote the empty substitution, and let $\mathrm{Sub}_{\mathcal{L}}$ denote the set of all substitutions for $\mathcal{L}$.

Update Models. An update model for a finite set of agents $\mathcal{A}$ and a language $\mathcal{L}$ is a quadruple $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$, where E is a finite non-empty set of events, R is a family of accessibility relations, one for every agent $a \in \mathcal{A}$, pre associates every event with a precondition, and sub associates every event with a substitution for $\mathcal{L}$:

$$
\begin{array}{lcl}
\mathsf{E} & = & \{\mathsf{e}_0, \ldots, \mathsf{e}_{n-1}\} \\
\mathsf{R} & = & \{\mathsf{R}(a) \subseteq \mathsf{E} \times \mathsf{E} \mid a \in \mathcal{A}\} \\
\mathsf{pre} & = & \{\mathsf{pre}(\mathsf{e}) \in \mathcal{L} \mid \mathsf{e} \in \mathsf{E}\} \\
\mathsf{sub} & = & \{\mathsf{sub}(\mathsf{e}) \in \mathrm{Sub}_{\mathcal{L}} \mid \mathsf{e} \in \mathsf{E}\}
\end{array}
$$

If $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$ is an update model and $\mathsf{e} \in \mathsf{E}$, then (U, e) is called a *pointed* update model, and e is called its *actual* event.

Executability. Let $M = (W, R, V)$ be an epistemic model for $\mathcal{A}$ and $\mathcal{P}$, and let $w \in W$. Let $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$ be an update model for $\mathcal{A}$ and any language $\mathcal{L}$ that can be interpreted in epistemic models for $\mathcal{A}$ and $\mathcal{P}$. Then an event $\mathsf{e} \in \mathsf{E}$ is called *executable* in (M, w) if and only if $M, w \vDash \mathsf{pre}(\mathsf{e})$.

Execution. Let M, w, and U be as above, and let $\mathsf{e} \in \mathsf{E}$ be executable in (M, w). Then the result of *executing* (U, e) in (M, w) is the epistemic model $M \otimes \mathsf{U} = (W', R', V')$ where:

$$
\begin{array}{lcl}
W' & = & \{(w, \mathsf{e}) \in W \times \mathsf{E} \mid \mathcal{M}, w \vDash \mathsf{pre}(\mathsf{e})\} \\
R'(a) & = & \{((w, \mathsf{e}), (w', \mathsf{e}')) \mid (w, w') \in R(a) \text{ and } (\mathsf{e}, \mathsf{e}') \in \mathsf{R}(a)\} \\
V'(p) & = & \{(w, \mathsf{e}) \in W' \mid M, w \vDash \mathsf{sub}(\mathsf{e})(p)\}
\end{array}
$$

and which has (w, e) as its actual world.

2.3 The Logic of Communication and Change

Language. The basic language LCC and the extended language $\mathsf{LCC}^{\cap}$ are obtained from PDL and $\mathsf{PDL}^{\cap}$, respectively, by adding a clause for update execution: if (U, e) is a pointed update model, then $[\mathsf{U}, \mathsf{e}]\varphi$ is a formula expressing that φ must be the case after successful execution of (U, e).

Semantics. The semantics of LCC and $\mathsf{LCC}^{\cap}$ is obtained from the semantics of PDL and $\mathsf{PDL}^{\cap}$, respectively, by adding the following clause for update execution:

$$[[\ [\mathsf{U},\mathsf{e}]\varphi\ ]]^M = \{w \in W \mid M, w \vDash \mathsf{pre}(\mathsf{e}) \text{ implies } (w,\mathsf{e}) \in [[\varphi]]^{M\otimes\mathsf{U}}\}$$

2.4 Expressive Power and Reduction Axioms

A truth-preserving translation from formulas in LCC to formulas in PDL is provided in [42]. The existence of this translation establishes that these two languages are equally expressive. We will show how to extend this translation to one from $\mathsf{LCC}^{\cap}$ to $\mathsf{PDL}^{\cap}$.

The main challenge is to translate formulas of the form $[\mathsf{U},\mathsf{e}_i][\pi]\varphi$ to equivalent formulas of the form $\bigwedge_{j=0}^{n-1}[T_{ij}^{\mathsf{U}}(\pi)][\mathsf{U},\mathsf{e}_j]\varphi$, where the T_{ij}^{U} operators transform every π-path in $M \otimes \mathsf{U}$ that starts in (w,e_i) and ends in (v,e_j) to a corresponding path from w to v in the original epistemic model M. We define the T_{ij}^{U} transformers as in [42], with an extra clause for intersection modalities:

$$\begin{array}{rcl}
T_{ij}^{\mathsf{U}}(a) & = & \begin{cases} ?\mathsf{pre}(\mathsf{e}_i); a & \text{if } (\mathsf{e}_i,\mathsf{e}_j) \in \mathsf{R}(a) \\ ?\bot & \text{otherwise} \end{cases} \\
T_{ij}^{\mathsf{U}}(?\phi) & = & \begin{cases} ?(\mathsf{pre}(\mathsf{e}_i) \wedge [\mathsf{U},\mathsf{e}_i]\phi) & \text{if } i = j, \\ ?\bot & \text{otherwise} \end{cases} \\
T_{ij}^{\mathsf{U}}(\pi_1;\pi_2) & = & \bigcup_{k=0}^{n-1} (T_{ik}^{\mathsf{U}}(\pi_1); T_{kj}^{\mathsf{U}}(\pi_2)) \\
T_{ij}^{\mathsf{U}}(\pi_1 \cup \pi_2) & = & T_{ij}^{\mathsf{U}}(\pi_1) \cup T_{ij}^{\mathsf{U}}(\pi_2) \\
T_{ij}^{\mathsf{U}}(\pi_1 \cap \pi_2) & = & T_{ij}^{\mathsf{U}}(\pi_1) \cap T_{ij}^{\mathsf{U}}(\pi_2) \\
T_{ij}^{\mathsf{U}}(\pi^*) & = & K_{ijn}^{\mathsf{U}}(\pi)
\end{array}$$

where $K_{ijn}^{\mathsf{U}}(\pi)$ is exactly as in [42], definition 26. To prove that:

$$M, w \vDash [\mathsf{U},\mathsf{e}_i][\pi]\varphi \quad \Leftrightarrow \quad M, w \vDash \bigwedge_{j=0}^{n-1}[T_{ij}^{\mathsf{U}}(\pi)][\mathsf{U},\mathsf{e}_j]\varphi$$

it is sufficient to show that:

$$(w,v) \in [[T_{ij}^{\mathsf{U}}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]]^M \quad \Leftrightarrow \quad ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [[\pi]]^{M\otimes\mathsf{U}}$$

This equivalence, to which we shall refer as (#), is established in [42] by induction on the complexity of π. We provide the additional induction step for intersection modalities (which is analogous to the induction step for union modalities in [42]). Assume that (#) holds for π_1 and π_2. Then:

$$
\begin{array}{ll}
 & (w,v) \in [[T_{ij}^{\mathsf{U}}(\pi_1 \cap \pi_2); ?\mathsf{pre}(\mathsf{e}_j)]]^M \\
\Leftrightarrow & (w,v) \in [[(T_{ij}^{\mathsf{U}}(\pi_1) \cap T_{ij}^{\mathsf{U}}(\pi_2)); ?\mathsf{pre}(\mathsf{e}_j)]]^M \\
\Leftrightarrow & (w,v) \in [[(T_{ij}^{\mathsf{U}}(\pi_1); ?\mathsf{pre}(\mathsf{e}_j)) \cap (T_{ij}^{\mathsf{U}}(\pi_2); ?\mathsf{pre}(\mathsf{e}_j))]]^M \\
\Leftrightarrow & (w,v) \in [[T_{ij}^{\mathsf{U}}(\pi_1); ?\mathsf{pre}(\mathsf{e}_j)]]^M \text{ and } (w,v) \in [[T_{ij}^{\mathsf{U}}(\pi_2); ?\mathsf{pre}(\mathsf{e}_j)]]^M \\
\Leftrightarrow \text{(IH)} & ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [[\pi_1]]^{M \otimes \mathsf{U}} \text{ and } ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [[\pi_2]]^{M \otimes \mathsf{U}} \\
\Leftrightarrow & ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [[\pi_1 \cap \pi_2]]^{M \otimes \mathsf{U}}
\end{array}
$$

This establishes that $\mathsf{LCC}^{\cap}$ and $\mathsf{PDL}^{\cap}$ are equally expressive. Moreover, it leads to a sound and complete proof system for $\mathsf{LCC}^{\cap}$ consisting of the proof system for $\mathsf{PDL}^{\cap}$ provided in [2] plus a number of so-called *reduction axioms*. The reduction axioms for $\mathsf{LCC}^{\cap}$ are exactly the same axioms that can be found in [42] for LCC:

$$
\begin{array}{lll}
[\mathsf{U},\mathsf{e}]\top & \leftrightarrow & \top \\
[\mathsf{U},\mathsf{e}]p & \leftrightarrow & (\mathsf{pre}(\mathsf{e}) \rightarrow \mathsf{sub}(\mathsf{e})(p)) \\
[\mathsf{U},\mathsf{e}]\neg\varphi & \leftrightarrow & (\mathsf{pre}(\mathsf{e}) \rightarrow \neg[\mathsf{U},\mathsf{e}]\varphi) \\
[\mathsf{U},\mathsf{e}](\varphi_1 \wedge \varphi_2) & \leftrightarrow & ([\mathsf{U},\mathsf{e}]\varphi_1 \wedge [\mathsf{U},\mathsf{e}]\varphi_2) \\
[\mathsf{U},\mathsf{e}_i][\pi]\varphi & \leftrightarrow & \bigwedge_{j=0}^{n-1}[T_{ij}^{\mathsf{U}}(\pi)][\mathsf{U},\mathsf{e}_j]\varphi
\end{array}
$$

2.5 Specific Communication Types

Some specific types of communication, such as public announcements and private messages to subgroups, are of special interest in dynamic epistemic logic. Every specific communication type corresponds to a specific kind of update model, which, in turn, gives rise to specific reduction axioms, in which the general T_{ij}^{U} operators do not appear anymore. Explicit reduction axioms for common knowledge were derived in [42]. Here, we give explicit reduction axioms for distributed knowledge. We treat two major communication types, that will also play a role in chapter 4.

Public announcements. A public announcement that φ is represented by an update model $\varphi!$ consisting of just one event e_0 with precondition φ and with accessibility relation $\{(\mathsf{e}_0, \mathsf{e}_0)\}$ for all agents (see figure 2.1).
The reduction axiom that expresses the effect of $\varphi!$ on distributed knowledge

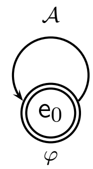


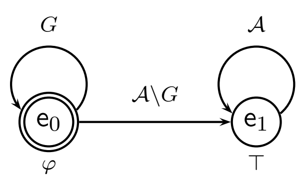
Figure 2.1: Public announcement that φ.

among B, can be derived as follows:

$$\begin{array}{rcl} [\varphi!]D_B\psi & \leftrightarrow & [\varphi!][\bigcap_{a\in B} a]\psi \\ & \leftrightarrow & [T_{00}^{\varphi!}(\bigcap_{a\in B} a)][\varphi!]\psi \\ & \leftrightarrow & [\bigcap_{a\in B} T_{00}^{\varphi!}(a)][\varphi!]\psi \\ & \leftrightarrow & [\bigcap_{a\in B}(?\mathsf{pre}(\mathsf{e}_0); a)][\varphi!]\psi \\ & \leftrightarrow & [\bigcap_{a\in B}(?\varphi; a)][\varphi!]\psi \\ & \leftrightarrow & [?\varphi; \bigcap_{a\in B} a][\varphi!]\psi \\ & \leftrightarrow & (\varphi \to [\bigcap_{a\in B} a][\varphi!]\psi) \\ & \leftrightarrow & (\varphi \to D_B[\varphi!]\psi) \end{array}$$

As was already conjectured in [42], this reduction axiom is just like the familiar reduction axiom for individual knowledge: $[\varphi!][a]\psi \leftrightarrow (\varphi \to [a][\varphi!]\psi)$.

Private messages to subgroups. A private message to subgroup G that φ is the case is represented by an update model consisting of two events e_0 and e_1 with precondition φ and $\top$, respectively. Intuitively, e_0 is the actual event, whereas e_1 is the event "nothing happens". Agents in G know that e_0 is taking place, while all the other agents think that nothing happens.



As in [42], we use CC_φ^G to denote this update model. The reduction axiom that expresses the effect of CC_φ^G on distributed knowledge among another subgroup B, can be derived as follows:

$$\begin{array}{rcl}
[CC^G_\varphi, \mathsf{e}_0]D_B\psi & \leftrightarrow & [CC^G_\varphi, \mathsf{e}_0][\bigcap_{a\in B} a]\psi \\
& \leftrightarrow & ([T_{00}^{CC^G_\varphi}(\bigcap_{a\in B} a)][CC^G_\varphi, \mathsf{e}_0]\psi \wedge \ldots \\
& & [T_{01}^{CC^G_\varphi}(\bigcap_{a\in B} a)][CC^G_\varphi, \mathsf{e}_1]\psi) \\
& \leftrightarrow & ([\bigcap_{a\in B}(T_{00}^{CC^G_\varphi}(a))][CC^G_\varphi, \mathsf{e}_0]\psi \wedge \ldots \\
& & [\bigcap_{a\in B}(T_{01}^{CC^G_\varphi}(a))][CC^G_\varphi, \mathsf{e}_1]\psi) \\
& \leftrightarrow & ([\bigcap_{a\in B\cap G}(?\varphi; a) \cap \bigcap_{a\in B\setminus G}(?\bot)][CC^G_\varphi, \mathsf{e}_0]\psi \wedge \ldots \\
& & [\bigcap_{a\in B\cap G}(?\bot) \cap \bigcap_{a\in B\setminus G}(?\varphi; a)][CC^G_\varphi, \mathsf{e}_1]\psi)
\end{array}$$

Now, we consider three cases:

1. Suppose $B \subseteq G$. Then the axiom reduces to:

$$\begin{array}{rcl}
[CC^G_\varphi, \mathsf{e}_0]D_B\psi & \leftrightarrow & [\bigcap_{a\in B}(?\varphi; a)][CC^G_\varphi, \mathsf{e}_0]\psi \\
& \leftrightarrow & [(?\varphi; \bigcap_{a\in B} a)][CC^G_\varphi, \mathsf{e}_0]\psi \\
& \leftrightarrow & (\varphi \to D_B[CC^G_\varphi, \mathsf{e}_0]\psi)
\end{array}$$

   If $B \subseteq G$, then all agents in B are aware of the announcement. As expected, the axiom says that *after* the announcement it is distributed knowledge among B that ψ is the case, if and only if *before* the announcement it was distributed knowledge among B that *should* the announcement be made, then ψ would become true.

2. Suppose $B \subseteq \mathcal{A} \setminus G$. Then the axiom reduces to:

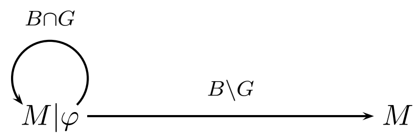
$$\begin{array}{rcl}
[CC^G_\varphi, \mathsf{e}_0]D_B\psi & \leftrightarrow & [\bigcap_{a\in B}(?\varphi; a)][CC^G_\varphi, \mathsf{e}_1]\psi \\
& \leftrightarrow & [(?\varphi; \bigcap_{a\in B} a)][CC^G_\varphi, \mathsf{e}_1]\psi \\
& \leftrightarrow & (\varphi \to D_B[CC^G_\varphi, \mathsf{e}_1]\psi) \\
& \leftrightarrow & (\varphi \to D_B\psi)
\end{array}$$

   If $B \subseteq \mathcal{A} \setminus G$, then none of the agents in B is aware of the announcement. As expected, the axiom says that *after* the announcement ψ is distributed knowledge among B if and only if this was already the case *before* the announcement.

3. Suppose $B \nsubseteq G$ and $B \nsubseteq \mathcal{A} \setminus G$. Then we get:

$$[CC^G_\varphi, \mathsf{e}_0]D_B\psi \quad \leftrightarrow \quad \top$$

   This result may seem surprising at first sight, but it has an intuitive and interesting explanation. Namely, the result of privately informing



some agents in B (those that belong to G) about φ, while keeping the others in the dark results in a true *separation* of beliefs.

As illustrated above, agents in $B \setminus G$ still believe they are in the original model M, while agents in $B \cap G$ have in fact shifted to a new model $M|\varphi$. As a consequence, there are *no* worlds in the overall resulting model that all agents in B consider possible. In other words, their collective information state is *inconsistent*, and therefore, every formula whatsoever has become distributed knowledge among them.

2.6 Summary

We have shown that the logic of communication and change can be extended so as to incorporate distributed knowledge and other intersection modalities. The extension is straightforward, and leads to intuitive reduction axioms for specific communication types.

Chapter 3

Communication Networks

We now move on to a more conceptual point. As mentioned before, distributed knowledge can be seen as the information that can be established by a group of agents through a particular kind of communication: every group member writes down everything he knows, and what follows from the accumulated facts is distributed knowledge. Other, arguably more interactive kinds of communication may also be considered. In particular, it would be interesting to analyze which information can be established by a group of agents relative to a restricted *communication network*. In this chapter, dynamic epistemic logic is used and extended to analyze communication relative to a communication network. Our aim is to outline a *general* framework, in which agents may have incomplete knowledge of the structure of the communication network, and in which the structure of the network itself is changeable.

3.1 Networks

In many situations, the extent to which agents are able to communicate with one another is restricted by their communication network. If agents communicate by phone, for instance, then each agent can only talk to *one* other agent at a time. If the communication medium is email, then whole groups of agents can be reached at once. But it may be the case that one email address is used by, say, a whole family of agents, so that it is impossible to send a private email message to one family member without the others finding out.

Networks. A communication network $N \subseteq \wp(\mathcal{A}) \times \wp(\mathcal{A})$ for a set of agents $\mathcal{A}$ is simply a set of communication *channels* (A, B) from one group of agents $A \subseteq \mathcal{A}$ to another group of agents $B \subseteq \mathcal{A}$.

3.2 A Logic of Communication Networks

We introduce a logic of communication networks, LCN, a subtle variation of the logic of communication and change, LCC, discussed in chapter 2.

Language. The language $\mathcal{L}^{\mathsf{LCN}}_{\mathcal{A},\mathcal{P}}$ is obtained from $\mathcal{L}^{\mathsf{LCC}}_{\mathcal{A},\mathcal{P}}$ by adding designated proposition letters $\mathrm{con}(A, B)$ for every $A, B \subseteq \mathcal{A}$, which are used to describe the structure of the communication network. Intuitively, $\mathrm{con}(A, B)$ means that the network contains a communication channel from A to B.

Update Models. Apart from general update models, we will consider two special kinds of update models. *Communication models* will be used to model communication relative to the communication network, and *reconfiguration models* will be used to model reconfigurations of the communication network. Both kinds of special update models will be discussed below.

Semantics. The semantics for LCN is exactly as the semantics for LCC. In particular, the clause for execution of an update model (U, e) in an epistemic model $M = (W, R, V)$ is as follows:

$$\mathcal{M}, w \vDash [\mathsf{U}, \mathsf{e}]\varphi \quad \text{iff} \quad M, w \vDash \mathsf{pre}(\mathsf{e}) \text{ implies } \mathcal{M} \otimes \mathsf{U}, (w, \mathsf{e}) \vDash \varphi$$

For general update models, $\mathcal{M} \otimes \mathsf{U}$ is defined exactly as for update models in LCC. The communication network plays no role in this case. For communication and reconfiguration models, $\mathcal{M} \otimes \mathsf{U}$ is defined in section 3.3 and 3.4, respectively. The proof system for LCN is just the same as that for LCC.

3.3 Communication

Communicative Events. Communicative events are distinguished from general information carrying events by the property of being initiated by and directed toward a particular group of agents. We define a communicative event e for $\mathcal{L}^{\mathsf{LCN}}_{\mathcal{A},\mathcal{P}}$ to be a triple $(A_{\mathsf{e}}, B_{\mathsf{e}}, \varphi_{\mathsf{e}})$ where:

- $A_{\mathsf{e}} \subseteq \mathcal{A}$ is the group of agents that sends e,

- $B_{\mathsf{e}} \subseteq \mathcal{A}$ is the group of agents that receives e,
- $\varphi_{\mathsf{e}} \in \mathcal{L}^{\mathsf{LCN}}_{\mathcal{A},\mathcal{P}}$ specifies the informational content of e.

Communication Models. A communication model for $\mathcal{L}^{\mathsf{LCN}}_{\mathcal{A},\mathcal{P}}$ is an update model $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$ for $\mathcal{L}^{\mathsf{LCN}}_{\mathcal{A},\mathcal{P}}$ where:

- E is a non-empty finite set of *communicative* events,
- R is a family of accessibility relations, one for every $a \in \mathcal{A}$,
- $\mathsf{pre}(\mathsf{e}) = \mathrm{con}(A_{\mathsf{e}}, B_{\mathsf{e}}) \wedge C_{A_{\mathsf{e}}}\varphi_{\mathsf{e}}$ for every $\mathsf{e} \in \mathsf{E}$,
- $\mathsf{sub}(\mathsf{e}) = \epsilon$ for every $\mathsf{e} \in \mathsf{E}$.

What distinguishes communication models from other update models is that (1) E contains only *communicative* events, (2) the preconditions associated with an event are determined by the specification of that event itself, and (3) all substitutions are empty.

Executability. A communicative event is executable if and only if its preconditions are satisfied. A first precondition for a communicative event e to take place is that the communication network supports communication from A_{e} to B_{e}. A second precondition is that φ_e is common knowledge among the sending group of agents. Together, $\mathsf{pre}(\mathsf{e}) = \mathrm{con}(A_{\mathsf{e}}, B_{\mathsf{e}}) \wedge C_{A_{\mathsf{e}}}\varphi_{\mathsf{e}}$.

Execution. Let $M = (W, R, V)$ be an epistemic model for $\mathcal{A}$ and $\mathcal{P}$, with $w \in W$. Let $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$ be a communication model for $\mathcal{L}^{\mathsf{LCN}}_{\mathcal{A},\mathcal{P}}$, and let $\mathsf{e} \in \mathsf{E}$ be executable in (M, w). Then the result of executing (U, e) in (M, w) is the epistemic model $M \otimes \mathsf{U} = (W', R', V')$ where:

$$\begin{array}{lcl} W' & = & \{(w,\mathsf{e}) \in W \times \mathsf{E} \mid \mathcal{M}, w \vDash \mathsf{pre}_{\mathsf{e}}\} \\ R'(a) & = & \{((w,\mathsf{e}),(w',\mathsf{e}')) \mid (w,w') \in R(a) \text{ and } (\mathsf{e},\mathsf{e}') \in \mathsf{R}(a)\} \\ V'((w,\mathsf{e})) & = & V(w) \end{array}$$

with (w, e) as its actual world. Thus, the execution of a communicative event model, unlike the execution of a general update model, *depends on* the communication network. However, like the execution of a general update model, it does not *affect* the communication network. The opposite is true for reconfiguration models, to be defined next.

3.4 Reconfiguration

Reconfigurations. A reconfiguration is a special kind of substitution, namely, one that maps all but a number of designated proposition letters of the form $\text{con}(A, B)$ to itself. A reconfiguration can be represented as a set of bindings:

$$\{\text{con}(A, B) \mapsto \varphi_{A,B} \mid A, B \in \mathcal{A}\}$$

Intuitively, $\text{con}(A, B) \mapsto \varphi_{A,B}$ means that a communication channel from A to B is added to the network in all worlds that satisfy $\varphi_{A,B}$. In particular, $\text{con}(A, B) \mapsto \top$ means that a channel from A to B is added everywhere, $\text{con}(A, B) \mapsto \bot$ means that the channel from A to B is removed everywhere, $\text{con}(A, B) \mapsto \neg\text{con}(A, B)$ means that a channel from A to B is added in worlds where it was not present, but removed in worlds where it *did* exist, and $\text{con}(A, B) \mapsto \text{con}(A, B)$ means that the channel from A to B is left untouched.

Reconfiguration Models. A reconfiguration model for $\mathcal{L}^{\mathsf{LCN}}_{\mathcal{A},\mathcal{P}}$ is an update model $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$ for $\mathcal{L}^{\mathsf{LCN}}_{\mathcal{A},\mathcal{P}}$ where:

- E is a non-empty finite set of *reconfigurations*,
- R is a family of accessibility relations, one for every $a \in \mathcal{A}$,
- pre associates a precondition $\mathsf{pre}(\mathsf{e}) \in \mathcal{L}^{\mathsf{LCN}}_{\mathcal{A},\mathcal{P}}$ with every $\mathsf{e} \in \mathsf{E}$,
- $\mathsf{sub}(\mathsf{e}) = \mathsf{e}$ for every $\mathsf{e} \in \mathsf{E}$.

What distinguishes a reconfiguration model from other update models is that (1) E contains only *reconfigurations*, and (2) the substitution associated with a reconfiguration is that reconfiguration itself.

Executability. A reconfiguration is executable if and only if its preconditions are satisfied. Preconditions for reconfigurations are like preconditions for events in general update models; there are no special restrictions.

Execution. Let $M = (W, R, V)$ be an epistemic model for $\mathcal{A}$ and $\mathcal{P}$, with $w \in W$. Let $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$ be a reconfiguration model for $\mathcal{L}^{\mathsf{LCN}}_{\mathcal{A},\mathcal{P}}$, and let $\mathsf{e} \in E$ be executable in $(\mathcal{M}, w)$. Then the result of executing (U, e) in $(\mathcal{M}, w)$ is the epistemic model $M \otimes \mathsf{U} = (W', R', V')$ where:

$$\begin{array}{rcl}
W' & = & \{(w, \mathsf{e}) \in W \times \mathsf{E} \mid \mathcal{M}, w \vDash \mathsf{pre}_{\mathsf{e}}\} \\
R'(a) & = & \{((w, \mathsf{e}), (w', \mathsf{e}')) \mid (w, w') \in R(a) \text{ and } (\mathsf{e}, \mathsf{e}') \in \mathsf{R}(a)\} \\
V'(p) & = & \{(w, \mathsf{e}) \in W' \mid M, w \vDash \mathsf{sub}(\mathsf{e})(p)\}
\end{array}$$

with (w, e) as its actual world. Thus, the execution of a reconfiguration model, unlike the execution of a general update model, *affects* the network.

3.5 Summary

The logic of communication networks we proposed only involved a minor extension of LCC. The main idea underlying our proposal is that *communicative* events are distinguished from other events by their inherent property of being initiated by and directed toward a particular group of agents.

The proposed framework seems to facilitate a very general analysis of communication relative to a partially observable, changeable network. The next chapter defines notions group knowledge relative to concrete instances of the general communication networks discussed here.

Chapter 4

Communicative Power

In this chapter, we introduce a new notion of group knowledge, called *potential knowledge*, and a more general notion called *communicative power*. Both notions are conceived of relative to a communication network.

Just as specific communication types, there are specific *network types* that seem to be of special interest. We propose a few major network types, and investigate potential knowledge relative to these network types.

We leave many issues open in this chapter. Our main objective, for now, is to introduce some relevant notions and sketch a plan for further research.

4.1 Communicative Power

We define a formula φ to be *potential knowledge* among a group of agents B if there is some finite sequence of communicative events, all initiated by subgroups of B, that establishes common knowledge of φ among B. Potential knowledge is probably the most straightforward dynamic notion of collective group knowledge.

A more general notion is what we call *communicative power*. A group B has the communicative power to establish φ if there is a finite sequence of communicative events, all initiated by subgroups of B, that yields φ. We write $P_B\varphi$ if B has the communicative power to establish φ. Potential knowledge of φ among B can then be expressed as $P_B C_B \varphi$. We may be interested in weaker forms of communicative power, such as the power to establish universal knowledge of φ. This can be expressed as $P_B E_B \varphi$. But we may also be interested in far more subtle forms of communicative power, such as the power of a group of agents B to share a certain piece of information φ without informing another agent a. This kind of communicative power plays a role in the *russian cards problem* [45]. It can be expressed as $P_B(C_B\varphi \wedge \neg[a]\varphi)$.

Axiomatizing communicative power does not seem to be straightforward. Moreover, the results in [24] cast serious doubt on its decidability. A possible way to avoid this would be to consider *bounded* versions of communicative power. Bounds on the number of communicative events, as well as bounds on the modal depth of the communicated formulas may be considered.

4.2 Specific Network Types

Many situations involve special types of communication networks. Every face-to-face group meeting, for example, involves a *public broadcast* network, in which every group member can only publicly address the rest of the group as a whole. Another type of network that is reminiscent of every day life is the one-to-one *private message passing* network provided by telephone operators. Other communication media, such as email, give rise to more complicated communication networks.

It may be possible and useful to further systematize this spectrum of specific network types. For now, we focus on a modest case study of potential knowledge in public broadcast and private message passing networks.

4.3 Potential Knowledge and Public Broadcast

Van Benthem [39] already showed that if a public broadcast network is available in finite, distinguishing models, then it is always possible for a group of agents B to communicate in such a way that the model is restricted to its *communicative core*: the set of worlds that every member of B considers possible. Clearly, everything that becomes common knowledge among B at some point of the communication, will be common knowledge in the core. Thus, for a finite, distinguishing model $M = (W, R, V)$ we have:

$$M, w \models P_B C_B \varphi \quad \text{iff} \quad \text{Core}_B(M, w) \models \varphi$$

where $\text{Core}_B(M, w)$ is the restriction of M to $\{v \in W \mid (w, v) \in \bigcap_{a \in B} R(a)\}$, again with w as its actual world. To reach the core, just *one* announcement per agent is needed. To see this, notice that for every world v in a distinguishing model $M = (W, R, V)$, there is a formula φ_v that is true in v and no-where else in M. So, if a pointed model (M, w) is finite every agent a can restrict it to her own information state by announcing $\bigvee_{v \in [M,w]_a} \varphi_v$. After all agents have done so, the model will be restricted to its communicative core.

This result still holds in infinite models, as long as $[M, w]_a$ is finite for all a and all w. If this is not the case, the core may not be reachable by a finite sequence of announcements. Still, it determines a useful upper bound for potential knowledge. We have:

$$M, w \vDash P_B C_B \varphi \quad \text{only if} \quad \text{Core}_B(M, w) \vDash \varphi$$

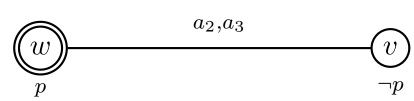
This upper bound can be approximated by ever more lengthy sequences.

4.4 Potential Knowledge and Private Messages

We consider the *three generals* example discussed in [39]. Notice that this example is *not* the same as the more familiar *two generals* scenario discussed, for instance, in [20]. The latter scenario is used to show that *unreliable* communication channels make it impossible to establish common knowledge. We show that private message passing networks yield similar restrictions.

Generals. Let $B = \{a_1, a_2, a_3\}$ be a group of three generals with armies on different hilltops, who plan to surprise their enemy in the plain below. Only a_1 has some piece of information p, which has to become common knowledge to facilitate a joint attack.

The initial model for this situation is:



The private message passing network only allows for one-to-one messages (hereafter simply called messages), which are represented by communication models (U, m) like the one depicted in figure 4.1, where m is a special kind of communicative event of the form $(s_\mathsf{m}, r_\mathsf{m}, \varphi_\mathsf{m})$, sent by $s_\mathsf{m} \in B$ and received by $r_\mathsf{m} \in B$. Notice that m completely determines (U, m). We will therefore simply write m instead of (U, m).
It can be shown, by induction on the number of messages that may be sent, that the generals do not have the communicative power to establish $C_B p$. It is clear that a single message can not establish $C_B p$. Now, suppose that there is no way for the generals to establish $C_B p$ with at most n messages. This means that every model that is obtained from the original model by

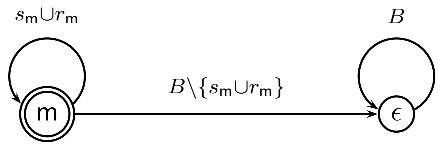


Figure 4.1: (U, m).

executing a sequence of n messages, still contains at least one B-path that leads from the actual world to a world where p is false. Let M be such a model, and let w be its actual world. Let $\mathsf{m} = (s_\mathsf{m}, r_\mathsf{m}, \varphi_\mathsf{m})$ be a message that is executable in (M, w), and let u_m be the only general who is *unaware* of m being sent ($u_\mathsf{m} \notin \{s_\mathsf{m}, r_\mathsf{m}\}$). The result of executing m in (M, w) is the model:

$$M|\mathsf{m} \xrightarrow{\quad u_\mathsf{m} \quad} M|\epsilon$$

where $M|\mathsf{m}$ is the restriction of M to worlds where φ_m is true, with accessibility relations for s_m and r_m only, while $M|\epsilon$ is like the original model M, the only difference being that every world w has been renamed to (w, ϵ). Every world in $M|\mathsf{m}$ is mistaken by u_m for the corresponding world in $M|\epsilon$. In particular, the *actual* world (w, m) is mistaken for (w, ϵ), and (w, ϵ) is still connected by some B-path to a world where p is false. So, also after $n + 1$ messages, $C_B p$ has not been established. By induction, then, we conclude that the generals do not have the communicative power to establish $C_B p$.

This example illustrates that the communication network between agents can have important limitative effects on the agents' communicative power. Private message passing networks are ubiquitous, but it would of course be interesting to consider other network types as well.

4.5 Knowledge over Time

Communicative power is about the possible manipulation of information by agents over time. In that respect, it fits into a whole range of questions that arise in a *temporal* perspective on epistemic logic. Communication *networks* should play a role in addressing these questions. Another dimension, which we did not mention here, is formed by communication *protocols*. These can be seen as subsets of the possible branches through a space of evolving epistemic models. In this perspective, the following kinds of issues arise:

- *Safety.* Does a protocol assure that nothing bad happens?
- *Eventuality.* Does a protocol assure that something good happens?
- *Power.* Has group B the communicative power to establish φ?
- *Limit behaviour.* Does iterated communication converge to a stable information state?

Safety and eventuality issues have been studied extensively in temporal logic, especially motivated by applications of this logic to system verification. Emerson [11] provides an overview of this work. Communicative power is related with game theoretical power notions and logical treatments thereof. Especially relevant are the formalization of group power in games by Pauly [27] and the links between game theory and dynamic epistemic logic described by van Benthem [38]. Limit behaviour of iterated communication is investigated in the work of Sadzik [33]. General treatments of temporal epistemic frameworks can be found in [13] and [26].

4.6 Summary

We introduced a new notion of group knowledge, called *potential knowledge*, and a more general notion called *communicative power*. Both notions were conceived of relative to a communication network. We related the potential knowledge of agents in public broadcast networks to the communicative core of their information models, and showed that private message passing networks impose important restrictions on agents' communicative power. In particular, we showed that common knowledge is unattainable in such networks. Finally, we suggested that communicative power issues naturally fall into a whole range of issues that arise in a temporal perspective on epistemic logic. Probably, they should be further investigated within this perspective.

Part II

Multi-context Systems

Chapter 5

Introduction

Some formalizations of distributed information and its dynamics take the *contextual* nature of information as their point of departure. The most notable frameworks of this kind are the propositional logic of context developed by McCarthy, Buvač and Mason [9, 21, 22], and the multi-context systems devised by Giunchiglia, Serafini, and Ghidini [17, 18, 19]. These frameworks have been compared from a technical viewpoint by Serafini and Bouquet [35] and from a more conceptual perspective by Benerecetti et.al. [5].

Multi-context systems, which are reviewed in section 5.1, describe the information available in a number of contexts (i.e., to a number of people, agents, databases, etc.) and specify the inter-contextual information flow. The so-called local model semantics defines a system to entail a certain piece of information in a certain context, if and only if that piece of information is acquired in that context, independently of how the information flow described by the system is accomplished.

In chapter 6, it is shown that the local model semantics of a multi-context system is completely determined by the information that is obtained when simulating the information flow specified by the system, in such a way that a *minimal* amount of information is deduced at each step of the simulation. We define an operator that suitably implements such a simulation, and thus determines the information entailed by the system. This operator constitutes a first constructive account of the local model semantics.

In chapter 7, it is observed that, in its original formulation, the multi-context system framework implicitly rests on the assumption that information flow is *deterministic*. In many situations, this is not a suitable assumption. In a multi-agent scenario, for example, upon establishing a certain piece of information, an agent may decide to pass this information on to either one of a group of other agents. His choice as to which agent he will inform could be made non-deterministically. Another typical situation in which information flow is inherently non-deterministic is when the information channels between different contexts are subject to temporary failure or unavailability. Consider the case of online repositories. If information is obtained in one repository, the protocol may be to pass this information on to any one of a number of associated "mirror repositories": if the communication channel with one of these is defective or temporarily unavailable, another one is tried, until a successful communication is established.

The local model semantics can easily be adapted to account for non-deterministic systems. However, if a system describes a non-deterministic information flow, then the minimal information entailed by the system cannot be determined unequivocally. We provide a way to generate from a non-deterministic system a number of deterministic systems, the semantics of which can be determined constructively, and which, together, completely determine the semantics of the original non-deterministic system.

In chapter 8, it is observed that in multi-context systems, new information is derived based on the *presence* of other information only. However, in many natural situations (concrete examples will be given below), new information is obtained due to a *lack* of other information. We propose a generalized framework to account for such situations. Non-monotonic reasoning techniques are applied to formulate a semantics for this framework.

5.1 Multi-Context Systems

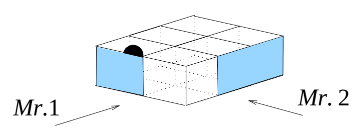


Figure 5.1: A magic box.

A simple illustration of the main intuitions underlying the multi-context system framework is provided by the situation depicted in figure 5.1. Two agents, Mr.1 and Mr.2, are looking at a box from different angles. The box is called magic, because neither Mr.1 nor Mr.2 can make out its depth. As some sections of the box are out of sight, both agents have partial information about the box. To express this information, Mr.1 only uses proposition letters l (there is a ball on the left) and r (there is a ball on the right), while Mr.2 also uses a third proposition letter c (there is a ball in the center).

In general, we consider a set of contexts I, and a language L_i for each context $i \in I$. Henceforward, we assume I and $\{L_i\}_{i \in I}$ to be fixed, unless specified otherwise. Moreover, we assume each L_i to be built over a finite set of proposition letters, using standard propositional connectives.

To state that the information expressed by a formula $\varphi \in L_i$ is established in context i we use so-called *labelled formulas* of the form $i : \varphi$ (if no ambiguity arises, we simply refer to labelled formulas as formulas, and we even use capital letters F, G, and H to denote labelled formulas, if the context label is irrelevant). A *rule* r is an expression of the form:

$$F \leftarrow G_1 \wedge \ldots \wedge G_n \tag{5.1}$$

where F and all G's are labelled formulas; F is called the consequence of r and is denoted by $cons(r)$; all G's are called premises of r and together make up the set $prem(r)$. Rules without premises are called *facts*. Rules with at least one premiss are called *bridge rules*. A *multi-context system* (system hereafter) is a finite set of rules. A fact describes information that is established in a certain context, independent of which information is obtained in other contexts. A bridge rule specifies which information is established in one context, if other pieces of information are obtained in different contexts. So a system can be seen as a specification of contextual information available a priori plus an inter-contextual information flow.

Example 5.1 *The scenario in figure 5.1 is modelled by the following system:*

$$\begin{array}{lcl} 1 : \neg r & \leftarrow & \\ 2 : l & \leftarrow & \\ 1 : l \vee r & \leftarrow & 2 : l \vee c \vee r \\ 2 : l \vee c \vee r & \leftarrow & 1 : l \vee r \end{array}$$

Mr.1 knows that there is no ball on the right, Mr.2 knows that there is a ball on the left, and if any agent gets to knows that there is a ball in the box, then he will inform the other agent about it.

A classical interpretations m of language L_i is called a *local model* of context i. A set of local models is called a *local information state*. Intuitively, every local model in a local information state represents a "possible state of affairs". If a local information state contains exactly one local model, then it represents complete information. If it contains more than one local model, then it represents partial information: more than one state of affairs is considered possible. A *distributed information state* is a set of local information states, one for each context. In conformity with the literature, we will refer to distributed information states as *chains*.

Example 5.2 *The situation in figure 5.1, in which Mr.1 knows that there is no ball on the right but does not know whether there is a ball on the left, is represented by a chain whose first component* $\{\{l, \neg r\}, \{\neg l, \neg r\}\}$ *contains two local models. As such, the chain reflects Mr.1's uncertainty about the left section of the box.*

A chain c *satisfies* a labelled formula $i : \varphi$ (denoted $c \models i : \varphi$) if and only if all local models in its i^{th} component classically satisfy φ. A rule r is *applicable* with respect to a chain c if and only if c satisfies every premiss of r. Notice that facts are applicable with respect to any chain. A chain c *complies with* a rule r, if and only if, whenever r is applicable with respect to c, then c satisfies r's consequence. We call c a *solution chain* of a system S if and only if it complies with every rule in S. A formula F is *true* in S (denoted $S \models F$) if and only if every solution chain of S satisfies F.

For convenience, we introduce some auxiliary terminology and notation. Let $\mathbf{C}$ denote the set of all chains. Notice that, as each L_i is assumed to be built over a finite set of proposition letters, $\mathbf{C}$ is assumed to be finite. Let $c^{\perp}$ denote the chain containing every local model of every context ($c^{\perp}$ does not satisfy any non-tautological expression); let $c^{\top}$ denote the chain containing no local models at all ($c^{\top}$ satisfies all expressions). If C is a set of chains, then the component-wise union (intersection) of C is the chain, whose i^{th} component consists of all local models that are in the i^{th} component of some (every) chain in C. If c and c' are chains, then $c \setminus c'$ denotes the chain, whose i^{th} component consists of all local models that are in c_i but not in c'_i. Finally, let us sometimes say that a local model m is (not) in c, when we actually mean that m is (not) in some (any) component c_i of c.

Chapter 6

Minimality

In this chapter, we show that the semantics of a multi-context system is completely determined by its least informative solution chain, and provide a way to compute this least informative solution chain.

6.1 Minimal Solution Chain Semantics

We can order chains according to the amount of information they convey. Intuitively, the more local models a chain component contains, the more possibilities it permits, so the less informative it is. Formally, we say that c is *less informative* than c' ($c \preceq c'$), if for every i we have $c_i \supseteq c'_i$. If, moreover, for at least one i we have $c_i \supset c'_i$, then we say that c is *strictly less informative* than c' ($c \prec c'$). Note that $c^{\perp}$ is strictly less informative than any other chain, and $c^{\top}$ is strictly more informative than any other chain.

Lemma 6.1 *If $c \preceq c'$, then any formula satisfied by c is also satisfied by c'.*

Proof. Suppose $c \models i : \phi$. Then, $m \models \phi$ for every $m \in c_i$. As c'_i is contained in c_i, we also have $m' \models \phi$ for every $m' \in c'_i$. So $c' \models i : \phi$. □

We say that c is *minimal* among a set of chains C, if c is in C and no other chain c' in C is strictly less informative than c. In particular, we say that c is a *minimal solution chain* of a system S, if it is minimal among the set of all solution chains of S. The main result of this section, theorem 6.5, establishes that every system has a unique minimal solution chain. To prove this result, it suffices to show that the component-wise union of all solution chains of a system S is less informative than all solution chains of S (lemma 6.2) and in fact itself a solution chain of S (lemma 6.4).

Lemma 6.2 *Let C be a set of chains. Let c^u denote the component-wise union of all chains in C. Then c^u is less informative than any chain in C.*

Proof. Let c' be a chain in C. Then for every i, every local model m in c'_i is also in c^u_i. So $c^u_i \supseteq c'_i$, and thus $c^u \preceq c'$. □

Lemma 6.3 *Let C be a set of chains and let c^u denote the component-wise union of all chains in C. Then a formula is satisfied by c^u if and only if it is satisfied by every chain in C.*

Proof.
($\Rightarrow$) Follows directly from lemma 6.2 and lemma 6.1.
($\Leftarrow$) Suppose all chains in C satisfy $i : \phi$. Then all local models in the i^{th} component of every chain in C must satisfy ϕ. These are exactly the local models that make up the i^{th} component of c^u. So c^u also satisfies $i : \phi$. □

Lemma 6.4 *The set of all solution chains of a system S is closed under component-wise union. I.e., if C is a set of solution chains of S, then the component-wise union c^u of all chains in C is again a solution chain of S.*

Proof. Let C be a set of solution chains of S. Let c^u be the component-wise union of C. Let r be an arbitrary rule in S. Then all c' in C comply with r. Suppose, toward a contradiction, that c^u does not comply with r, i.e., c^u satisfies all of r's premises, but does not satisfy r's consequence. By lemma 6.3 all c' in C satisfy all of r's premises, and therefore, by assumption, they all satisfy r's consequence as well. But then, again by lemma 6.3, c^u must also satisfy r's consequence, which contradicts the assumption that c^u does not comply with r. So c^u must comply with r, and as r was arbitrary, c^u must be a solution chain of S. □

Theorem 6.5 *Every system S has a unique minimal solution chain c_S.*

Proof. Every system has at least one solution chain, namely $c^\top$. Now, let S be a system and let C_S be the set of all its solution chains. Then, by lemma 6.4, the component-wise union c_S of C_S is itself in C_S. Moreover, by lemma 6.2, c_S is less informative than any other chain in C_S. So c_S is minimal among C_S and, moreover, any chain c' in C_S which is minimal among C_S, must be equal to c_S. In other words, c_S is the unique minimal solution chain of S. □

Theorem 6.6 *The semantics of a system S is completely determined by its unique minimal solution chain c_S. That is, for any formula F we have:*

$$S \models F \quad \Leftrightarrow \quad c_S \models F$$

Proof. Let S be a system and let F be a formula. Then F is true in S if and only if F is satisfied by all solution chains of S. By lemma 6.3, this is the case if and only if F is satisfied by the component-wise union of all solution chains of S. By the proof of theorem 6.5 this union constitutes the minimal solution chain c_S of S. □

Theorem 6.5 and 6.6 establish that, to answer queries about a system S, it is no longer necessary to compute all solution chains of S; we only need to consider the system's minimal solution chain c_S.

6.2 Computing the Minimal Solution Chain

Recall that a system S can be thought of as a specification of inter-contextual information flow. It turns out that the minimal solution chain of S can be characterized as the $\preceq$-least fixpoint of an operator $\mathbf{T}_S$, which, intuitively, simulates the information flow specified by S.

Let $S^*(c)$ denote the set of rules in S that are applicable w.r.t. c. Then:

$$\mathbf{T}_S(c) = c \setminus \{m \mid \exists r \in S^*(c) : m \not\models cons(r)\} \tag{6.1}$$

For every rule r in S that is applicable w.r.t. c, $\mathbf{T}_S$ removes from c all local models that do not satisfy $cons(r)$. Intuitively, this corresponds to augmenting c with the information expressed by $cons(r)$. In this sense, $\mathbf{T}_S$ simulates the information flow described by S. Clearly, $\mathbf{T}_S(c)$ is obtained from c only by *removing* local models from it. As a result, $\mathbf{T}_S(c)$ is always more informative than c.

Lemma 6.7 *For every chain c and every system S: $c \preceq \mathbf{T}_S(c)$.* □

We now prove that, starting with the least informative chain $c^{\perp}$, $\mathbf{T}_S$ will reach its $\preceq$-least fixpoint after finitely many iterations, and that this $\preceq$-least fixpoint coincides with the minimal solution chain of S. The first result is typically established using Tarski's fixpoint theorem [36]. In order to apply this theorem, we first need to show that $(\mathbf{C}, \preceq)$ forms a complete lattice, and that $\mathbf{T}_S$ is monotone and continuous with respect to $\preceq$.

Lemma 6.8 *$(\mathbf{C}, \preceq)$ forms a complete lattice.*

Proof. We should prove that every finite subset of $\mathbf{C}$ has both a greatest lower bound and a least upper bound in $\mathbf{C}$. Let C be a subset of $\mathbf{C}$ (note that $\mathbf{C}$ is finite, so C must be finite as well). Let c^u (c^i) denote the componentwise union (intersection) of all chains in C. Then, by lemma 6.2, c^u is a lower bound of C. Now consider a chain c', such that $c^u \prec c'$. For this to be the case, there must be a local model m, which is in c^u but not in c'. But then m must also be in some chain c^m in C, which makes $c' \preceq c^m$ impossible. So c' cannot be a lower bound of C, which implies that c^u is the greatest lower bound of C. Analogously, it is shown that c^i is least upper bound of C. □

Lemma 6.9 *$\mathbf{T}_S$ is monotone with respect to $\preceq$.*

Proof. Let c and c' be any two chains such that $c \preceq c'$. We need to prove that $\mathbf{T}_S(c) \preceq \mathbf{T}_S(c')$. Suppose, toward a contradiction that this is not the case. Then there is a local model m that belongs to $\mathbf{T}_S(c')$ but not to $\mathbf{T}_S(c)$. Clearly, m must already be present in c', and therefore also in c. From the fact that m has been removed from c by $\mathbf{T}_S$ it follows that there must be a rule r in S such that c satisfies $prem(r)$, whereas m does not satisfy $cons(r)$. But then, by lemma 6.1, c' must also satisfy $prem(r)$, so $\mathbf{T}_S$ should have removed m from c' as well. We conclude that $\mathbf{T}_S(c) \preceq \mathbf{T}_S(c')$. So $\mathbf{T}_S$ is monotone with respect to $\preceq$. □

Lemma 6.10 *$\mathbf{T}_S$ is continuous with respect to $\preceq$.*

Proof. Let $c^0 \preceq c^1 \preceq c^2 \preceq \ldots$ be an infinite sequence of chains, each of which contains more information than all preceding ones. We need to prove that $\mathbf{T}_S(\bigcup_{n=0}^{\infty} c^n) = \bigcup_{n=0}^{\infty} \mathbf{T}_S(c^n)$. As $\mathbf{C}$ is finite, $\{c^0, c^1, c^2, \ldots\}$ must have a maximum c^m in $\mathbf{C}$. So $\mathbf{T}_S(\bigcup_{n=0}^{\infty} c^n) = \mathbf{T}_S(c^m) = \bigcup_{n=0}^{\infty} \mathbf{T}_S(c^n)$. □

Theorem 6.11 *$\boldsymbol{T}_S$ has a $\preceq$-least fixpoint, which is obtained after a finite number of consecutive applications of $\boldsymbol{T}_S$ to $c^\perp$.*

Proof. Lemmas 6.8, 6.9, 6.10, and Tarski's fixpoint theorem [36]. □

Lemma 6.12 *Let c be a chain and let S be a system. Then c is a fixpoint of $\boldsymbol{T}_S$ if and only if c is a solution chain of S.*

Proof. A chain c is a fixpoint of $\mathbf{T}_S$ if and only if for every rule r in S, c satisfies $cons(r)$ whenever c satisfies $prem(r)$. This is the case if and only if c is a solution chain of S. □

Theorem 6.13 *Let S be a system. Then the minimal solution chain c_S of S coincides with the $\preceq$-least fixpoint of $\boldsymbol{T}_S$.*

Proof. Follows directly from lemma 6.12. □

From theorems 6.11 and 6.13 we conclude that the minimal solution chain c_S of a system S is obtained by a finite number of applications of $\mathbf{T}_S$ to the least informative chain $c^\perp$. But we can even prove a slightly stronger result:

Theorem 6.14 *Let S be a system and let $|S|$ denote the number of* bridge rules *in S. Then the minimal solution chain c_S of S is obtained by at most $|S|+1$ consecutive applications of $\boldsymbol{T}_S$ to $c^\perp$.*

Proof. Let c be a chain and let S be a system. Notice that $\mathbf{T}_S(c)$ is a fixpoint of $\mathbf{T}_S$ if and only if $S^*(\mathbf{T}_S(c))$ coincides with $S^*(c)$. Lemmas 6.1 and 6.7 imply that, in any case, $S^*(\mathbf{T}_S(c)) \supseteq S^*(c)$. In other words, during each iteration of $\mathbf{T}_S$ some (possibly zero) rules are added to S^*. In the case that S^* remains unaltered, $\mathbf{T}_S$ must have reached a fixpoint. Now we observe that during the first application of $\mathbf{T}_S$ (to $c^\perp$) all facts in S are added to S^*. Clearly, after that, $\mathbf{T}_S$ can be applied at most $|S|$ times before a fixpoint is reached. □

In fact, a slightly more involved, but essentially equivalent procedure was introduced for rather different reasons in [31]. This procedure was shown to have worst-case time complexity $O(|S|^2 \times 2^M)$, where M is the maximum number of propositional variables in either one of the contexts involved in S. The greater part of a typical computation is taken up by propositional reasoning within individual contexts, which itself requires exponential time in the worst case.

Example 6.1 *Consider the system S given in example 5.1. Applying $\boldsymbol{T}_S$ to $c^{\perp}$ establishes the facts given by the first two rules of the system. But then Mr.2 knows that there is a ball in the box, so the next application of $\boldsymbol{T}_S$ simulates the information flow specified by the third rule of the system: Mr.2 informs Mr.1 of the presence of the ball. The resulting chain is left unaltered by any further application of $\boldsymbol{T}_S$, and therefore constitutes the minimal solution chain of S. The fact that this chain satisfies the formula $1 : l$ reflects, as desired, that Mr.1 has come to know that there is a ball in the left section of the box.*

6.3 Summary

We have shown that the local model semantics of a multi-context system is completely determined by the minimal solution chain of that system. This solution chain represents the information that is obtained when the information flow specified by the system is simulated in such a way that a *minimal* amount of information is deduced at each step of the simulation. We defined an operator that suitably implements such a minimal simulation, and thus determines the minimal solution chain of the system.

Chapter 7

Non-Determinism

The original formulation of multi-context systems implicitly rests on the assumption that information flow is *deterministic*. However, there are many natural situations in which information flow is inherently non-deterministic. To model such situations, this chapter develops a non-deterministic extension of the multi-context systems framework.

7.1 Non-Deterministic Multi-Context Systems

Let us first consider an example of a situation in which information flows non-deterministically.

Example 7.1 *Adriano is on holiday after having submitted his final school exams. He has promised to call his father or his mother in case his teacher lets him know that he has passed his exams. This situation can be modeled by a system S consisting of the following rule:*

$$m : p \vee f : p \leftarrow a : p$$

Notice that Adriano may be conceived of as an agent in a multi-agent system, who non-deterministically decides which other agents to inform when acquiring novel information. Alternatively, Adriano's parents may be conceived of as mirror repositories of information about Adriano's well-being (assuming that they tell each other everything they come to know about Adriano). Typical telephonic connections may be broken or temporarily unavailable. Analogous to the situation sketched in the introduction, Adriano will try to reach his parents, until at least one of them is informed. In general, we would like to consider systems in which rules r are of the form:

$$F_1 \vee \ldots \vee F_m \leftarrow G_1 \wedge \ldots \wedge G_n \tag{7.1}$$

where all F's and G's are labeled formulas; all F's are called consequences of r and together form the set $cons(r)$; and as before, all G's are called premises of r and together constitute the set $prem(r)$. A rule does not necessarily have any premises ($n \geq 0$), but always has at least one consequence ($m \geq 1$). We call a rule deterministic if it has only one consequence, and non-deterministic otherwise. We call finite sets of possibly non-deterministic rules *non-deterministic multi-context systems* (non-deterministic systems for short). Systems which consist of deterministic rules only, are from now on referred to as deterministic systems.

A chain c complies with a non-deterministic rule r if and only if, whenever r is applicable w.r.t. c, *at least one of* its consequences is satisfied by c. A chain is a solution chain of S if and only if it complies with all rules in S. A formula F is true in S, $S \models F$, if and only if F is satisfied by all solution chains of S.

Observation 7.1 *Let S be a non-deterministic system, let c' and c'' be two solution chains of S, and let c be the component-wise union of c' and c''. Then it is* not *generally the case that c is again a solution chain of S. Therefore, S does not generally have a unique minimal solution chain.*

Example 7.2 *Suppose Adriano is informed that he passed his exams. The resulting system S is given by the following rules:*

$$\begin{array}{rcl} a:p & \leftarrow & \\ m:p \vee f:p & \leftarrow & a:p \end{array}$$

This system has two minimal solution chains:

$$\begin{array}{rcl} c^m & = & \{\{p\}_m, \quad \{p, \neg p\}_f, \quad \{p\}_a\} \\ c^f & = & \{\{p, \neg p\}_m, \quad \{p\}_f, \quad \{p\}_a\} \end{array}$$

whose component-wise union $\{\{p, \neg p\}_m, \quad \{p, \neg p\}_f, \quad \{p\}_a\}$ is not a solution chain of S.

Theorem 7.2 *The semantics of a non-deterministic system S is completely determined by the set C_S of all its minimal solution chains. For any formula F we have:*

$$S \models F \quad \Leftrightarrow \quad \forall c \in C_S : c \models F$$

Proof. Let S be a non-deterministic system and let F be a formula. Then F is true in S if and only if F is satisfied by every solution chain of S. Clearly, if F is satisfied by every solution chain of S, then it must in particular be satisfied by every minimal solution chain of S. Moreover, every solution chain of S is an extension of some minimal solution chain of S, which implies, by lemma 6.1, that F is satisfied by all minimal solution chains of S only if F is satisfied by all solution chains of S. □

Theorem 7.2 establishes that the meaning of a non-deterministic system S is completely determined by the set C_S of all its minimal solution chains. Next, we show how to compute C_S using the method outlined in chapter 6.

7.2 Minimal Solution Chain Semantics

Inspired by an idea originally developed for disjunctive databases [34], we generate from a non-deterministic system S a number of deterministic systems $S_1, S_2, \ldots, S_n$, in such a way that the minimal solution chains of S are among the minimal solution chains of $S_1, S_2, \ldots, S_n$ (note that each S_i has a unique minimal solution chain which can be computed as outlined in section 6). Hereto, we introduce the notion of a *generated system*. Let S be a non-deterministic system and let r be a rule in S. Then we say that a deterministic rule r' is *generated by* r if and only if $cons(r') \in cons(r)$ and $prem(r') = prem(r)$. We say that a system S' is *generated by* S if and only if it is obtained from S by replacing each rule r in S by some rule r' generated by r. Notice that, indeed, a generated system is always deterministic, and that any non-deterministic system S generates at most $\prod_{r \in S} |cons(r)|$ different deterministic systems.

Example 7.3 *The non-deterministic system from example 7.2 generates two deterministic systems:*

$$\begin{array}{lll} a:p & \leftarrow & \\ m:p & \leftarrow & a:p \end{array} \qquad\qquad \begin{array}{lll} a:p & \leftarrow & \\ f:p & \leftarrow & a:p \end{array}$$

The only system generated by a deterministic system is that system itself.

Lemma 7.3 *A chain c is a solution chain of a non-deterministic system S if and only if it is a solution chain of some system S' generated by S.*

Proof.
($\Rightarrow$) Suppose c is a solution chain of S. Then c complies with every rule in S. For every rule r in S, if c complies with r, then there must be a rule r' generated by r such that c complies with r' as well. Let S' be the system $\{r' \mid r \in S\}$. Then c is a solution chain of S'.
($\Leftarrow$) Suppose c is a solution chain of a system S' generated by S. Then c complies with every rule in S'. Every rule r in S has generated some rule r' in S', and clearly, if c complies with r' then it must also comply with r. So c is a solution chain of S. □

We call a chain c a *potential solution chain* of S if and only if c is a minimal solution chain of some system S' generated by S.

Lemma 7.4 *Every minimal solution chain of a system S is also a potential solution chain of S.*

Proof. Suppose c is a minimal solution chain of S. Then, by lemma 7.3, c is a solution chain of some system S' generated by S. Let c' be the minimal solution chain of S'. Then c must be an extension of c'. By lemma 7.3 c' must be a solution chain of S. But then, as c is a minimal solution chain of S, c' must be equal to c. So c is a minimal solution chain of S', and therefore a potential solution chain of S. □

Observation 7.5 *It is* not *generally the case that a potential solution chain of S is also a minimal solution chain of S.*

Example 7.4 *Suppose Adriano's teacher also called Adriano's mother to tell her the good news. The resulting system S is given by the following rules:*

$$\begin{array}{rcl} a:p & \leftarrow & \\ m:p & \leftarrow & \\ m:p \vee f:p & \leftarrow & a:p \end{array}$$

This system has two potential solution chains:

$$\begin{array}{rcl} c^m & = & \{\{p\}_m, \quad \{p, \neg p\}_f, \quad \{p\}_a\} \\ c^{mf} & = & \{\{p\}_m, \quad \{p\}_f, \quad \{p\}_a\} \end{array}$$

But as c^{mf} extends c^m only the latter is a minimal solution chain of S.

We call c an *essential solution chain* of S if and only if c is minimal among all potential solution chains of S.

Theorem 7.6 *A chain is a minimal solution chain of S if and only if it is an essential solution chain of S.*

Proof.
($\Rightarrow$) Suppose c is a minimal solution chain of S. Then, by lemma 7.4, c is a potential solution chain of S. If c is minimal among all potential solution chains of S, then, per definition, it is essential. Now, toward a contradiction, suppose that c is *not* minimal among all potential solution chains of S. Then there must be another potential solution chain c' of S, such that $c' \prec c$. But, by lemma 7.3, c' must also be a solution chain of S, which contradicts the assumption that c is a minimal solution chain of S.
($\Leftarrow$) Suppose c is an essential solution chain of S. Furthermore, toward a contradiction, suppose that c is *not* a minimal solution chain of S. Then there must be a minimal solution chain c' of S, such that $c' \prec c$. By lemma 7.4, c' is a potential solution chain of S. But this contradicts the assumption that c is minimal among all potential solution chains of S. □

Theorem 7.6 establishes that, in order to compute the meaning of a non-deterministic system S it suffices to compute the meaning of all deterministic systems generated by S. This can be done re-using the method developed in chapter 6. Given that S generates at most $\prod_{r \in S} |cons(r)|$ different systems, and that computing the meaning of each of these systems takes at most time $O(|S|^2 \times 2^M)$, we conclude that, in the worst case, computing the meaning of S takes time $O(\prod_{r \in S} |cons(r)| \times |S|^2 \times 2^M)$.

7.3 Summary

We observed that multi-context systems can only be used to describe deterministic information flow. We sketched a number of situations in which information flow is non-deterministic. We extended the framework in order to account for non-deterministic information flow and showed how to express the semantics of a non-deterministic system in terms of the semantics of a number of associated, deterministic systems, which can be computed using the machinery developed in chapter 6.

Chapter 8

Absent Information

Multi-context systems can only be used to model information flow in which new information is established based on the *presence* of other information. There are many natural situations in which information is obtained as a result of the *absence* of other information. This chapter develops an extension of the present framework in order to deal with such situations.

8.1 Normal Multi-Context Systems

Let us first consider two typical situations in which new information is obtained as a result of the absence of other information.

Example 8.1 (Integration) *Let d_1, d_2 be two databases, and let d_3 be a third database, which integrates the information established in d_1 and d_2, respectively. Any piece of information that is established by d_1 and not refuted by d_2 (or vice versa) is included in d_3:*

$$
\begin{aligned}
3 : \varphi &\leftarrow 1 : \varphi \wedge \mathbf{not}\ 2 : \neg\varphi \\
3 : \varphi &\leftarrow 2 : \varphi \wedge \mathbf{not}\ 1 : \neg\varphi
\end{aligned}
$$

Example 8.2 (Trust) *Let d_1, d_2, and d_3 be as in example 8.1. It would be natural for d_3 to regard d_1 as more trustworthy than d_2 (or vice versa). In this case, any piece of information that is established in d_1 is automatically included in d_3, but information obtained in d_2 is only included in d_3 if it is not refuted by d_1:*

$$
\begin{aligned}
3 : \varphi &\leftarrow 1 : \varphi \\
3 : \varphi &\leftarrow 2 : \varphi \wedge \mathbf{not}\ 1 : \neg\varphi
\end{aligned}
$$

In general, to model situations in which new information is obtained based on the absence of other information we need rules r of the form[1]:

$$F \leftarrow G_1 \wedge \ldots \wedge G_m \wedge \textbf{not } H_1 \wedge \ldots \wedge \textbf{not } H_n \tag{8.1}$$

where F, all G's, and all H's are labeled formulas. As before, F is called the consequence of r ($cons(r)$). $G_1, \ldots, G_m$ are called *positive premises* of r and together constitute the set $prem^+(r)$. $H_1, \ldots, H_n$ are called *negative premises* of r and make up the set $prem^-(r)$. A rule does not necessarily have any premises ($m, n \geq 0$). In analogy with commonplace terminology in deductive database and logic programming theory, we call such rules *normal* rules, and finite sets of them *normal multi-context systems* (normal systems for short). If a rule only has positive premises, we call it a *positive* rule. Note that a system, which consists of positive rules only conforms with the original definition of multi-context systems. From now on we call such systems *positive* systems.

Our aim is to generalize the result obtained chapter 6, i.e. to define the semantics of a normal system S in terms of a single *canonical* chain c_S of S, such that, whenever S is a positive system, c_S coincides with the minimal solution chain of S.

A first naive attempt would be to say that a chain c complies with a normal rule r if and only if it satisfies r's consequence, whenever it satisfies every positive premise of r and does not satisfy any negative premise of r. The (minimal) solution chains of a normal system S can then be defined as for positive systems. However, as the following example shows, a normal system does not generally have a unique minimal solution chain, and worse, minimal solution chains of a normal system do not generally correspond with the intended meaning of that system.

Example 8.3 *Let a system S be given by the following rule:*

$$1 : p \quad \leftarrow \quad \textbf{not } 2 : q$$

Then S has two minimal solution chains:

$$\begin{aligned} c^p &= \left\{ \begin{bmatrix} \{p\} \\ \phantom{\{\neg p\}} \end{bmatrix}_1 \begin{bmatrix} \{q\} \\ \{\neg q\} \end{bmatrix}_2 \right\} \\ c^q &= \left\{ \begin{bmatrix} \{p\} \\ \{\neg p\} \end{bmatrix}_1 \begin{bmatrix} \{q\} \\ \phantom{\{\neg q\}} \end{bmatrix}_2 \right\} \end{aligned}$$

[1] For now, we take deterministic systems as a starting point. The results in this section are *not* straightforwardly generalized to the case of non-deterministic systems.

Intuitively, S provides no ground for deriving q in context 2. Thus, p should be derived in context 1, and every "proper" canonical chain of S should satisfy $1 : p$. As c^q fails to do so, it should be rejected as such. How, then, should the canonical chain of a normal system be characterized?

Extensive research efforts have been involved with an analogous question in the setting of logic programming, when, in the late 80's / early 90's, a proper semantics for normal logic programs was sought. In motivating our characterization of canonical chains for normal multi-context systems, we will recall some important intuitions and adapt some crucial definitions that have resulted from these efforts.

A first desired property of canonical chains, first introduced in the setting of logic programming by Apt, Blair, and Walker [1] and Bidoit and Froidevaux [6], is termed *supportedness.* Intuitively, a chain c is a supported solution chain of a normal system S if and only if, whenever c satisfies a formula F, then S *provides an explanation for why this is so.*

Definition 8.1 *We call a chain c a supported solution chain of a normal system S if and only if, whenever c satisfies a formula F, then S contains a set R of rules, such that:*

- $\forall r \in R : \begin{cases} \forall G \in prem^+(r) : c \models G \\ \forall H \in prem^-(r) : c \not\models H \end{cases}$
- $\bigcup_{r \in R} cons(r) \models F$

Example 8.4 *In example 8.3, as desired, c^p is a supported solution chain of S, while c^q is not. But both c^p and c^q are supported solution chains of the following extension S' of S:*

$$\begin{aligned} 1 : p &\leftarrow \textbf{not } 2 : q \\ 2 : q &\leftarrow 2 : q \end{aligned}$$

Intuitively, c^p should be accepted as a canonical chain of S', but c^q should be rejected as such, because the explanation provided by S' for the fact that c^q satisfies $2 : q$ is *circular*, i.e., it relies on the very fact that c^q satisfies $2 : q$. So, in general, the concept of supportedness does not satisfactorily characterize the canonical chain of a normal system.

The notion of *well-supportedness*, first introduced for logic programs by Fages [12], refines the notion of supportedness to avoid the counter-intuitive result obtained in example 8.4. Intuitively, a chain c is a well-supported solution chain of a normal system S if and only if, whenever c satisfies a formula F, then S provides a *non-circular* explanation for why this is so.

Fages also proved this notion to be equivalent to the notion of *stability*, which had been defined somewhat earlier by Gelfond and Lifschitz [15]. The results obtained in chapter 6 pave the way for a straightforward adaptation of the notion of stability to our present setting.

Definition 8.2 *Let c be a chain and S a normal system. Define:*

$$\begin{array}{rcl} S'(c) & = & \{r \in S \mid \forall H \in prem^-(r) : c \not\models H\} \\ S''(c) & = & pos(S'(c)) \end{array}$$

where $pos(S'(c))$ is obtained from $S'(c)$ by removing all negative premises from its rules. Then, c is a stable solution chain of S, iff it is the unique minimal solution chain of $S''(c)$.

Intuitively, a solution chain c of a system S is stable if, whenever the information represented by c is assumed, then the information flow specified by S reproduces exactly c. Namely, if c is assumed to contain valid information, then any rule in S, one of whose negative premises is satisfied by c, is certainly not applicable. Negative premises which are *not* satisfied by c can be removed from the remaining rules, because they do not have any influence on whether those rules are applicable or not. Thus, S can be reduced to $S''(c)$, and c is stable if and only if it corresponds exactly to the meaning of $S''(c)$, i.e., by Theorem 6.6, to its minimal solution chain.

Example 8.5 *In example 8.4, as desired, c^p is a stable solution chain of S', while c^q is not.*

For many systems, stability suitably characterizes a unique canonical chain. There are still some special cases, however, in which it fails to do so. We give some typical examples.

Example 8.6 *Both c^p and c^q from example 8.3 are stable solution chains of the system given by the following rules:*

$$\begin{array}{rcl} 1:p & \leftarrow & \textbf{not}\ 2:q \\ 2:q & \leftarrow & \textbf{not}\ 1:p \end{array}$$

Example 8.7 *The following system does not have any stable solution chains.*

$$1:p \quad \leftarrow \quad \textbf{not}\ 1:p$$

In both cases we think it is most reasonable to conclude that no information is derived at all, i.e. to regard $c^{\perp}$ as the proper canonical chain.

Example 8.8 *The following system does not have any stable solution chains either.*

$$\begin{aligned} 1:p &\leftarrow \textbf{not } 1:p \\ 1:t &\leftarrow \textbf{not } 2:q \\ 2:r &\leftarrow 1:t \end{aligned}$$

Example 8.8 illustrates that, even if the rest of the system is unproblematic, one single rule (in this case the first one) can cause the system not to have any stable solution chain at all. In this case, t and r should be derived in context 1 and 2, respectively.

The *well-founded semantics*, first proposed for logic programs by van Gelder, Ross, and Schlipf [47] avoids the problems encountered in the above examples. The well-founded model of a program is defined as the least fixpoint of an operator, which, given an interpretation, determines the atoms that are necessarily true and those that are necessarily not true with respect to the program and the interpretation. It assigns *true* to the former set of atoms, and $false$ to the latter. As a result, more atoms may become necessarily true or necessarily not true. Corresponding truth values are assigned until a fixpoint is reached. All atoms that have not been assigned a definite truth value, are interpreted as *unknown*.

Our approach shares an important intuition with the well-founded semantics for logic programs, namely, that while constructing the canonical chain of a system, it is not only important to accumulate the information that *can certainly be derived* from the system, but also to keep track of information that *can certainly not be derived* from the system.

But the two approaches are also fundamentally different. The well-founded semantics constructs a 3-valued interpretation I, which is minimal with respect to a *truth order* $\sqsubseteq$ (i.e. $I \sqsubseteq I'$ iff I makes less atoms true and more atoms false than I'), whereas we seek a chain which is minimal with respect to an *information order* $\preceq$ (i.e. $c \preceq c'$ iff c makes less expressions either true or false than c'). This particularly results in a different treatment of expressions that are found *not* to be true. To regard these expressions as false, as the well-founded semantics does, would be to introduce redundant information. Instead, in our setting, such expressions should simply be recorded as not being derivable.

8.2 Anti-Chain Semantics

The canonical chain of a normal system S, henceforward denoted by c_S, is constructed by an iterative transformation of a datastructure $\langle c, a\rangle$, where:

- c is the "canonical chain under construction". Initially, $c = c^{\perp}$. Every transformation of c removes from it those local models that are found not to be in c_S. So at any phase of the construction of c_S, c contains those local models that are *possibly* in c_S, and as such represents the information that is *necessarily* conveyed by c_S.

- a is the "anti-chain". Initially, $c = c^{\top}$. Every transformation of a adds to it those local models that are found to be in c_S. So at any phase of the construction of c_S, a contains those local models that are *necessarily* in c_S, and as such represents the information that is *possibly* conveyed by c_S.

Observation 8.3 *By construction, we have $c \preceq c_S \preceq a$. Therefore, by lemma 6.1, for any formula F:*

$$\begin{aligned} c \models F \quad &\Rightarrow \quad c_S \models F \\ a \not\models F \quad &\Rightarrow \quad c_S \not\models F \end{aligned}$$

□

We call a chain-anti-chain pair $\langle c, a\rangle$ *less evolved* than another such pair $\langle c', a'\rangle$ (denoted as $\langle c, a\rangle \leq \langle c', a'\rangle$) if and only if c is less informative than c' and a is more informative than a'. If, moreover, c is strictly less informative than c' or a is strictly more informative than a', then we say that $\langle c, a\rangle$ is strictly less evolved than $\langle c', a'\rangle$. We say that $\langle c, a\rangle$ is *minimal* among a set $\mathcal{CA}$ of chain-anti-chain pairs, if and only if $\langle c, a\rangle \in \mathcal{CA}$ and no other chain-anti-chain pair $\langle c', a'\rangle$ in $\mathcal{CA}$ is strictly less evolved than $\langle c, a\rangle$. Notice that, if $\langle c, a\rangle$ is minimal among $\mathcal{CA}$, then c is minimal among $\{c \mid \langle c, a\rangle \in \mathcal{CA}\}$.

Given a certain chain-anti-chain pair $\langle c, a\rangle$, the intended transformation $\mathbf{\Psi}_S$ first determines which rules in S will (not) be applicable w.r.t. c_S, and then refines $\langle c, a\rangle$ accordingly. The canonical chain c_S of S will be characterized as the first component of the $\leq$-least fixpoint of $\mathbf{\Psi}_S$.

We first specify how $\mathbf{\Psi}_S$ determines which rules will (not) be applicable w.r.t. c_S. Let $\langle c, a\rangle$ and a rule r in S be given. If r has a positive premise G, which is satisfied by c, then G will also be satisfied by c_S. On the other hand, if r has a negative premiss H, which is *not* satisfied by a, then H will not be satisfied by c_S either. So if all positive premises of r are satisfied by c and all negative premises of r are not satisfied by a, then r will be applicable with respect to c_S:

$$S^+(c, a) \quad = \quad \left\{ r \in S \;\middle|\; \begin{array}{c} \forall G \in \mathit{prem}^+(r) : c \models G \\ \text{and} \\ \forall H \in \mathit{prem}^-(r) : a \not\models H \end{array} \right\}$$

If r has a positive premise G, which is not satisfied by a, then G will not be satisfied by c_S either. If r has a negative premise H, which is satisfied by c, then H will be satisfied by c_S as well. In both cases r will certainly not be applicable with respect to c_S:

$$S^-(c, a) \quad = \quad \left\{ r \in S \;\middle|\; \begin{array}{c} \exists G \in \mathit{prem}^+(r) : a \not\models G \\ \text{or} \\ \exists H \in \mathit{prem}^-(r) : c \models H \end{array} \right\}$$

For convenience, we write:

$$S^\sim(c, a) = S \setminus S^-(c, a)$$

Think of $S^\sim(c, a)$ as the set of rules that is *possibly* applicable with respect to c_S, and notice that $S^+(c, a) \subseteq S^\sim(c, a)$, whenever $c \preceq a$, and that $S^+(c, a) = S^\sim(c, a)$, if $c = a$.

Lemma 8.4 *If S is a normal system and $\langle c, a\rangle$ and $\langle c', a'\rangle$ are two chain-anti-chain pairs s.t. $\langle c, a\rangle \leq \langle c', a'\rangle$, then we have:*

1. $S^+(c, a) \subseteq S^+(c', a')$
2. $S^-(c, a) \subseteq S^-(c', a')$
3. $S^\sim(c, a) \supseteq S^\sim(c', a')$

Proof. Suppose that $\langle c, a\rangle \leq \langle c', a'\rangle$. Then, by definition, $c \preceq c'$ and $a' \preceq a$. Let r be a rule in S. For the first statement, suppose that $r \in S^+(c, a)$. Then c satisfies all of r's positive premises, and a does not satisfy any of r's negative premises. By lemma 6.1, the same goes for c' and a', respectively, which implies that $r \in S^+(c', a')$. The second statement is proven analogously; the third follows directly from the second. □

Next, we specify how $\mathbf{\Psi}_S$ refines $\langle c, a\rangle$, based on $S^+(c, a)$ and $S^\sim(c, a)$. Every local model $m \in c_i$ that does not satisfy the consequence of a rule in

$S^+(c, a)$ should certainly not be in c_S and is therefore removed from c. On the other hand, every local model $m \in c_i$ that satisfies the consequences of every rule in $S^\sim(c, a)$ should certainly be in c_S (S provides no ground for removing it) and is therefore added to a.

$$\mathbf{\Psi}_S(\langle c, a\rangle) \quad = \quad \langle \mathbf{\Psi}_S^c(\langle c, a\rangle), \mathbf{\Psi}_S^a(\langle c, a\rangle)\rangle$$

where:

$$\begin{aligned} \mathbf{\Psi}_S^c(\langle c, a\rangle) &= c \setminus \left\{m \mid \exists r \in S^+(c, a) : m \not\models \mathit{cons}(r)\right\} \\ \mathbf{\Psi}_S^a(\langle c, a\rangle) &= a \cup \{m \mid \forall r \in S^\sim(c, a) : m \models \mathit{cons}(r)\} \end{aligned}$$

Notice that $\mathbf{\Psi}_S^c$ only *removes* local models from c, whereas $\mathbf{\Psi}_S^a$ only *adds* local models to a.

We now prove that, starting with $\langle c^\perp, c^\top\rangle$, $\mathbf{\Psi}_S$ reaches its $\leq$-least fixpoint after finitely many iterations. To apply Tarski's fixpoint theorem, we first need to show that $(\mathbf{C} \times \mathbf{C}, \leq)$ forms a complete lattice, and that $\mathbf{\Psi}_S$ is monotone and continuous with respect to $\leq$.

Lemma 8.5 *$(\boldsymbol{C} \times \boldsymbol{C}, \leq)$ forms a complete lattice.*

Proof. Let $\mathcal{CA}$ be a set of chain-anti-chain pairs. Let c^u and a^u (c^i and a^i) denote the component-wise union (intersection) of all chains c and a, respectively, such that $\langle c, a\rangle$ is in $\mathcal{CA}$. Then $\langle c^u, a^i\rangle$ is the greatest lower bound of $\mathcal{CA}$ and $\langle c^i, a^u\rangle$ is the least upper bound of $\mathcal{CA}$. The proof of this statement is completely analogous to that of lemma 6.8. □

Lemma 8.6 *$\mathbf{\Psi}_S$ is monotone with respect to $\leq$, that is, for every normal system S, $\langle c, a\rangle \leq \langle c', a'\rangle$ implies $\mathbf{\Psi}_S(\langle c, a\rangle) \leq \mathbf{\Psi}_S(\langle c', a'\rangle)$.*

Proof. Let S be a normal system and let $\langle c, a\rangle$ and $\langle c', a'\rangle$ be any two chain-anti-chain pairs such that $\langle c, a\rangle \leq \langle c', a'\rangle$. Then, by definition, $c \preceq c'$ and $a' \preceq a$. We need to prove that $\mathbf{\Psi}_S(\langle c, a\rangle) \leq \mathbf{\Psi}_S(\langle c', a'\rangle)$. Suppose, toward a contradiction, that this is not the case. Then $\mathbf{\Psi}_S^c(\langle c, a\rangle) \not\preceq \mathbf{\Psi}_S^c(\langle c', a'\rangle)$ or $\mathbf{\Psi}_S^a(\langle c', a'\rangle) \not\preceq \mathbf{\Psi}_S^a(\langle c, a\rangle)$. We consider both possibilities.

$\mathbf{\Psi}_S^c(\langle c, a\rangle) \not\preceq \mathbf{\Psi}_S^c(\langle c', a'\rangle)$

In this case there must be a local model m which is contained in $\mathbf{\Psi}_S^c(\langle c', a'\rangle)$ but not in $\mathbf{\Psi}_S^c(\langle c, a\rangle)$. In the process of applying $\mathbf{\Psi}_S^c$ to $\langle c', a'\rangle$ local models may be removed from c', but no local models are added to it. So m must already be present in c'. As $c \preceq c'$, m must also be in c, thus it must have been removed from c in the process of applying $\mathbf{\Psi}^c_S$ to $\langle c, a\rangle$. It follows that there must be a rule r in $S^+(c,a)$, such that $m \not\models cons(r)$. From the fact that $\langle c, a\rangle \leq \langle c', a'\rangle$, by lemma 8.4, it follows that $S^+(c,a) \subseteq S^+(c',a')$. So r is also in $S^+(c',a')$ and m should be removed from c' in the process of applying $\mathbf{\Psi}^c_S$ to $\langle c', a'\rangle$ as well. This contradicts our earlier conclusion that $m \in \mathbf{\Psi}^c_S(\langle c', a'\rangle)$.

$\mathbf{\Psi}^a_S(\langle c', a'\rangle) \not\preceq \mathbf{\Psi}^a_S(\langle c, a\rangle)$

In this case there must be a local model m which is contained in $\mathbf{\Psi}^a_S(\langle c, a\rangle)$ but not in $\mathbf{\Psi}^a_S(\langle c', a'\rangle)$. For $m \notin \mathbf{\Psi}^a_S(\langle c', a'\rangle)$ to hold, there must be a rule r in $S^\sim(c',a')$ such that $m \not\models cons(r)$. As $\langle c, a\rangle \leq \langle c', a'\rangle$, by lemma 8.4, we have $S^\sim(c',a') \subseteq S^\sim(c,a)$. So r must be in $S^\sim(c,a)$ as well, and therefore m cannot be in $\mathbf{\Psi}^a_S(\langle c, a\rangle)$. This contradicts our earlier conclusion that $m \in \mathbf{\Psi}^a_S(\langle c, a\rangle)$.

It follows that $\mathbf{\Psi}_S(\langle c, a\rangle) \leq \mathbf{\Psi}_S(\langle c', a'\rangle)$, as desired. □

Lemma 8.7 $\mathbf{\Psi}_S$ *is continuous with respect to* $\leq$.

Proof. Let $\langle c_0, a_0\rangle \leq \langle c_1, a_1\rangle \leq \langle c_2, a_2\rangle \leq \ldots$ be an infinite sequence of chain-anti-chain pairs, each of which is more evolved than all preceding ones. We need to prove that $\mathbf{\Psi}_S(\bigcup_{n=0}^{\infty}\langle c_n, a_n\rangle) = \bigcup_{n=0}^{\infty}\mathbf{\Psi}_S(\langle c_n, a_n\rangle)$. As $\mathbf{C}\times\mathbf{C}$ is finite, $\{\langle c_0, a_0\rangle, \langle c_1, a_1\rangle, \langle c_2, a_2\rangle, \ldots\}$ must have a maximum $\langle c_m, a_m\rangle$ in $\mathbf{C}\times\mathbf{C}$. So $\mathbf{\Psi}_S(\bigcup_{n=0}^{\infty}\langle c_n, a_n\rangle) = \mathbf{\Psi}_S(\langle c_m, a_m\rangle) = \bigcup_{n=0}^{\infty}\mathbf{\Psi}_S(\langle c_n, a_n\rangle)$. □

Theorem 8.8 $\mathbf{\Psi}_S$ *has a* $\leq$*-least fixpoint, which is obtained after finitely many iterations of* $\mathbf{\Psi}_S$*, starting with* $\langle c^\perp, c^\top\rangle$.

Proof. Lemmas 8.5, 8.6, 8.7, and Tarski's fixpoint theorem [36]. □

Definition 8.9 *Let* S *be a normal system, and let* $\langle c_S, a_S\rangle$ *be the* $\leq$*-least fixpoint of* $\mathbf{\Psi}_S$*. We define* c_S *to be the canonical chain of* S*, and we define the semantics of* S *to be completely determined by* c_S*. That is, for every formula* F*:*

$$S \models F \quad \equiv \quad c_S \models F$$

□

A bound on the number of iterations needed by $\mathbf{\Psi}_S$ to reach its $\leq$-least fixpoint can be formulated in terms of the number of bridge rules in S.

Theorem 8.10 *Let S be a normal system and let $|S|$ denote the number of* bridge rules *in S. Then, starting with $\langle c^\perp, c^\top \rangle$, $\mathbf{\Psi}_S$ will reach its $\leq$-least fixpoint after at most $|S|+1$ iterations.*

Proof. A chain-anti-chain pair $\langle c, a \rangle$ is a fixpoint of $\mathbf{\Psi}_S$ if and only if $S^+(c,a) = S^+(\mathbf{\Psi}_S(\langle c, a \rangle))$ and $S^-(c,a) = S^-(\mathbf{\Psi}_S(\langle c, a \rangle)))$. During the first application of $\mathbf{\Psi}_S$ (to $\langle c^\perp, c^\top \rangle$), all facts in S are added to S^+ and all bridge rules in S are added to $S^\sim$. Each further application of $\mathbf{\Psi}_S$ either leads to a fixpoint or to the addition of at least one bridge rule to S^+ or S^-. Once a bridge rule is added to S^+ or S^-, it will not be removed again in any further iteration of $\mathbf{\Psi}_S$. It follows that $\mathbf{\Psi}_S$ must reach its $\leq$-least fixpoint after at most $|S|+1$ iterations. □

The next theorem shows that definition 8.9 is a proper generalization of the local model semantics for positive systems.

Theorem 8.11 *Let S be a positive system. Then its canonical chain coincides with its minimal solution chain.*

Proof. If S is a positive system, then for every pair $\langle c, a \rangle$, $S^+(\langle c, a \rangle)$ coincides with $S^*(c)$ and therefore $\mathbf{\Psi}_S^c(\langle c, a \rangle)$ is independent of a. As a consequence, $\langle c_S, a_S \rangle$ is the $\leq$-least fixpoint of $\mathbf{\Psi}_S$, for some anti-chain a_S, if and only if c_S is the $\preceq$-least fixpoint of $\mathbf{T}_S$. □

The canonical chain of a system S, and other fixpoints of $\mathbf{\Psi}_S$, are intimately related to the stable solution chains of S.

Lemma 8.12 *If c is a stable solution chain of a normal system S, then $\langle c, c \rangle$ is a fixpoint of $\mathbf{\Psi}_S$.*

Proof. Recall that:

$$
\begin{aligned}
S'(c) &= \{r \in S \mid \forall H \in prem^-(r) : c \not\models H\} \\
&= \left\{ r \in S \;\middle|\; \begin{array}{l} \exists G \in prem^+(r) : c \not\models G \\ \forall H \in prem^-(r) : c \not\models H \end{array} \right\} \\
&\quad \underbrace{\cup \left\{ r \in S \;\middle|\; \begin{array}{l} \forall G \in prem^+(r) : c \models G \\ \forall H \in prem^-(r) : c \not\models H \end{array} \right\}}_{S^+(c,c)}
\end{aligned}
\tag{8.2}
$$

and that c is a stable solution chain of S, only if it is the minimal solution chain of $S''(c) = pos(S'(c))$. Furthermore, observe that for every c, $S^+(c, c) = S^\sim(c, c)$, which implies that $\langle c, c\rangle$ is a fixpoint of $\mathbf{\Psi}_S$ if and only if:

$$c = \{m \mid \forall r \in S^+(c, c) : m \models cons(r)\} \tag{8.3}$$

Suppose that $\langle c, c\rangle$ is *not* a fixpoint of $\mathbf{\Psi}_S$. There are two possibilities to consider:

$\exists m \in c : \exists r \in S^+(c, c) : m \not\models cons(r)$

In this case, c is not a solution chain of $S^+(c, c)$, and therefore not a solution chain of $S''(c)$.

$\exists m \notin c : \forall r \in S^+(c, c) : m \models cons(r)$

Suppose c is a solution chain of $S''(c)$. Then every rule in $S''(c)$ is such that c either satisfies its consequence, or does not satisfy at least one of its premises. Let c' be the chain obtained from c by adding m to it, and let r be a rule in $S''(c)$. If c satisfies r's consequence, then, by definition of m, c' does so as well. If c does not satisfy a premiss G of r, then, by lemma 6.1 and the fact that $c' \preceq c$, c' does not satisfy G either. It follows that c' is a solution chain of $S''(c)$ as well. So c is not the *minimal* solution chain of $S''(c)$.

In both cases, as desired, c is not a stable solution chain of S. □

Theorem 8.13 *Let S be a normal system, let $\langle c_S, a_S\rangle$ be the $\leq$-least fixpoint of $\mathbf{\Psi}_S$, and let c_{stable} be a stable solution chain of S. Then $c_S \preceq c_{stable} \preceq a_S$.*

Proof. Suppose that $c_S \not\preceq c_{stable}$ or that $c_{stable} \not\preceq a_S$. Then $\langle c_S, a_S\rangle \not\leq \langle c_{stable}, c_{stable}\rangle$, while $\langle c_{stable}, c_{stable}\rangle$, by lemma 8.12, is a fixpoint of $\mathbf{\Psi}_S$. This contradicts the assumption that $\langle c_S, a_S\rangle$ is the $\leq$-least fixpoint of $\mathbf{\Psi}_S$. □

Lemma 8.14 *Let S be a normal system. If $\langle c, c\rangle$ is the $\leq$-least fixpoint of $\mathbf{\Psi}_S$, then c is a stable solution chain of S.*

Proof. Recall, as in the proof of lemma 8.12, that c is a stable solution chain of S only if it is a minimal solution chain of $S''(c)$, that $S''(c)$ can be expressed in terms of $S^+(c, c)$ as in equation (8.2), and that $\langle c, c\rangle$ is a fixpoint of $\mathbf{\Psi}_S$ if and only if c satisfies condition (8.3).

Suppose that $\langle c, c\rangle$ is the $\leq$-least fixpoint of $\mathbf{\Psi}_S$. We first show that c is a solution chain of $S''(c)$. Observe that every rule in $S''(c) \setminus pos(S^+(c, c))$ is such that at least one of its premises is not satisfied by c, and that every rule in $pos(S^+(c, c))$ is such that c satisfies its consequence. It follows that c is a solution chain of $S''(c)$.

Next, we show that, if c is not a *minimal* solution chain of $S''(c)$, then $\langle c, c\rangle$ cannot be the $\leq$-least fixpoint of $\mathbf{\Psi}_S$.

Claim 8.15 *If c' is a solution chain of $S''(c)$ such that $c' \prec c$, then there is an a' such that $\langle c', a'\rangle$ is a fixpoint of $\mathbf{\Psi}_S$.*

By construction, $\langle c, c\rangle \not\leq \langle c', a'\rangle$, so from claim 8.15 it would follow directly that $\langle c, c\rangle$ is not the $\leq$-least fixpoint of $\mathbf{\Psi}_S$.

To prove claim 8.15 we show that, if a is such that $c \preceq a$, then $\mathbf{\Psi}_S^c(\langle c', a\rangle) = c'$. To see this, first notice that, as $\langle c', a\rangle \leq \langle c, c\rangle$, by lemma 8.4, we have $S^+(c', a) \subseteq S^+(c, c)$. Now, let r be a rule in $S^+(c, c)$. As c' complies with all rules in $pos(S^+(c, c))$, c' must satisfy $cons(r)$, whenever it satisfies all positive premises of r. Every rule r' in $S^+(c', a)$ is also in $S^+(c, c)$ and is such that c' satisfies all its positive premises. Therefore, c' must satisfy $cons(r')$ as well. This means that no local models are removed from c' in the process of applying $\mathbf{\Psi}_S^c$ to $\langle c', a\rangle$. In other words: $\mathbf{\Psi}_S^c(\langle c', a\rangle) = c'$.

For any a we have $\mathbf{\Psi}_S^a(\langle c', a\rangle) \preceq a$ (no local models will be removed from a in the process of applying $\mathbf{\Psi}_S^a$ to $\langle c', a\rangle$). So in particular, the sequence:

$$c, \mathbf{\Psi}_S^a(\langle c', c\rangle), \mathbf{\Psi}_S^a(\mathbf{\Psi}_S^a(\langle c', c\rangle)), \ldots$$

is a descending sequence with respect to $\preceq$, bounded by $c^\perp$. This means that after applying $\mathbf{\Psi}_S$ finitely many times to $\langle c', c\rangle$ a fixpoint $\langle c', a'\rangle$ of $\mathbf{\Psi}_S$ will be reached, which proves claim 8.15.

We conclude that c is a minimal solution chain of $S''(c)$, and therefore a stable solution chain of S. □

Theorem 8.16 *Let S be a normal system and let $\langle c_S, a_S \rangle$ be the $\leq$-least fixpoint of $\mathbf{\Psi}_S$. If c_S and a_S coincide, then c_S is the unique stable solution chain of S.*

Proof. Stability of c_S is established by lemma 8.14; uniqueness follows from theorem 8.13. □

Finally, we remark that, in our view, all the examples presented above are suitably dealt with by the present analysis. We treat one of them explicitly.

Example 8.9 *Let S be the system from example 8.8. Then:*

$$c_S \quad = \quad \left\{ \begin{bmatrix} \{p, t\} \\ \{\neg p, t\} \end{bmatrix}_1 \quad \begin{bmatrix} \{q, r\} \\ \{\neg q, r\} \end{bmatrix}_2 \right\}$$

As desired, no information is derived about p and q, while t and r are indeed established in context 1 and 2, respectively.

8.3 Summary

Multi-context systems can only be used to model information flow in which new information is obtained due to the presence of other information. We provided some examples of situations in which the establishment of new information depends not only on the presence, but also on the absence of other information. We presented a generalized framework to model such situations. We applied non-monotonic reasoning techniques in defining a semantics for this framework.

Remark. Recently, Gerhard Brewka noticed that the anti-chain semantics for normal multi-context systems defined above has a significant drawback. The problem is related to a problem he and Georg Gottlob encountered in generalizing the well-founded semantics to default logic [8]. We will sketch the problem and a possible solution here, but the issue certainly calls for further investigation.

To see what goes wrong, consider the following system S:

$$\begin{array}{llll}(a) & 1:p & \leftarrow & \textbf{not}\ 1:\neg p \\ (b) & 1:\neg p & \leftarrow & \textbf{not}\ 1:p \\ (c) & 1:t & \leftarrow & \textbf{not}\ 1:q\end{array}$$

Intuitively, one would like $1 : t$ to be true in this system. But it is not satisfied by the canonical chain of the system. The problem is that rules (a) and (b) are both possibly applicable, but *no* local model can satisfy the consequences of both rules (p and $\neg p$). So no local model is added to the anti-chain, and the fact that $1 : q$ is not derivable will never be recognized.

The following seems to be a possible solution. Consider *multiple* anti-chains, one for each subsystem consisting of rules whose consequences do not enforce local inconsistencies. Then update each anti-chain relative to the set of rules that it is associated with. As an example, consider the system S above. There are two subsystems of S that consist of rules whose consequences do not enforce local inconsistencies:

$$\begin{array}{lll}S_1 & = & \{(a),(c)\} \\ S_2 & = & \{(b),(c)\}\end{array}$$

So we work with two anti-chains, a_1 and a_2. In general, let $\mathbf{a}$ denote the set of anti-chains of a system. The update of anti-chains, then, is as follows:

> Add to $a_n \in \mathbf{a}$ every local model that satisfies the consequence of every possibly applicable rule in the associated subsystem S_n.

Notice that before we had:

> Add to a every local model that satisfies the consequence of every possibly applicable rule in S.

In the above example we get:

$$\begin{array}{lll}a_1' & = & \{pqr, p\neg qr\} \\ a_2' & = & \{\neg pqr, \neg p\neg qr\}\end{array}$$

Next, we adjust the definition of $S^+(c, \mathbf{a})$, the set of rules that will definitely be applicable with respect to c_S. A rule is in $S^+(c, \mathbf{a})$ if and only if:

1. all its positive premises are satisfied by c.
2. none of its negative premises is satisfied by *any* anti-chain $a_n \in \mathbf{a}$. (instead of: "…by *the* anti-chain a".)

In the above example, rule (c) is in $S^+(c, \mathbf{a})$, which makes $1 : r$ true in S, as desired. This adapted semantics could be referred to as the *anti-chainset semantics* for normal multi-context systems.

Chapter 9

Conclusions

We conclude with a short summary of our main observations and results. In part one, we investigated the notion of distributed knowledge in epistemic and dynamic epistemic logic. Distributed knowledge is a standard notion in this framework, but nevertheless, there are some subtle unresolved issues concerning its semantics. Urged by these issues, we critically examined epistemic semantics, and proposed a rather dramatic simplification thereof. We argued that epistemic models are only acceptable if they are modally saturated and tight. Non-redundant models stand out as most natural, and we argued that epistemic semantics should be defined exclusively in terms of such models. This simplification resolves the problematic issues concerning distributed knowledge, and is, in our view, also of considerable independent interest.

Next, we showed that the logic of communication and change [42], the most general and most perspicuous dynamic epistemic logic to date, can be extended so as to incorporate distributed knowledge and other intersection modalities in a rather straightforward way. This extension could not be considered before, because a suitable notion of bisimulation for intersection modalities was not available. Restricting epistemic semantics to natural models resolves this technicality.

Then, we moved on to a more conceptual issue. We observed that distributed knowledge is a purely static notion of collective group knowledge. For a theory of communication, it should be of interest to consider more dynamic notions of collective group knowledge as well. The question becomes: which information can be established by a group through communication? To address this question, the communication network between a group of agents has to be taken into account. We showed that a logic of communication networks can be obtained as a slight extension of the logic of communication and change. We defined a general dynamic notion of collective group knowledge, called communicative power, and investigated a specific kind of communicative power, called potential knowledge, in public broadcast and private message passing networks.

In part two, we investigated the multi-context system formalism as a framework for representing distributed information and its dynamics. We observed that the semantics of a multi-context system is completely determined by the information that is obtained when simulating the information flow specified by the system, in such a way that a *minimal* amount of information is deduced at each step of the simulation. Based on this observation, we defined an operator that determines the information entailed by the system by suitably simulating the prescribed information flow.

Next we observed that the multi-context system framework implicitly rests on the assumption that information flow is *deterministic*. We sketched a number of situations, in which this assumption is not valid. We extended the framework in order to account for non-deterministic information flow, and provided a way to express the semantics of a non-deterministic system in terms of the semantics of a number of associated, deterministic systems.

Finally, we observed that in the multi-context framework, new information is deduced based on the *presence* of other information. We presented a generalized framework that accounts for situations in which new information can be derived based on the *absence* of other information as well.

The general background of this work is to use ideas from non-monotonic reasoning to obtain suitable semantics for extended multi-context systems. This connection opens up a whole range of questions for further research.

We have not mentioned the *relationship* between the two formalisms considered in part one and part two, respectively. At least two different *kinds* of information seem to be involved. In epistemic logic and in the singular contexts of a multi-context system, *information about a situation* is modeled as a *range* of possible alternatives for the actual situation.

In multi-context systems, another kind of information, namely the *information that one situation carries about other situations*, is modeled by rules that describe the *regularities* between different contexts in a system.

Van Benthem [41] recognizes these different concepts of information and explores their treatment in a single modal framework. We think this issue is very much worth further investigation.

Bibliography

[1] K. R. Apt, H. A. Blair, and A. Walker. Towards a theory of declarative knowledge. 1988.

[2] P. Balbiani and D. Vakarelov. PDL with intersection of programs: a complete axiomatization. *Journal of Applied Non-Classical Logics*, 13(2):231–267, 2003.

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

[4] A. Baltag, L. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Theoretical Aspects of Rationality and Knowledge (TARK-98)*, pages 43–56, 1998.

[5] M. Benerecetti, P. Bouquet, and C. Ghidini. Contextual reasoning distilled. *Journal of Experimental and Theoretical Artificial Intelligence*, 12(3):279–305, 2000.

[6] N. Bidoit and C. Froidevaux. General logical databases and programs: Default logic semantics and stratification. *Information and Computation*, 91:15–54, 1991.

[7] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[8] G. Brewka and G. Gottlob. Well-founded semantics for default logic. *Fundamenta Informaticae*, 31(3/4):221–236, 1997.

[9] S. Buvač and I. A. Mason. Propositional logic of context. In *National Conference on Artificial Intelligence (AAAI 93)*, 1993.

[10] R. Danecki. Nondeterministic propositional dynamic logic with intersection is decidable. In *Fifth Symposium on Computation Theory*, pages 34–53, 1984.

[11] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of theoretical computer science (vol. B): formal models and semantics*, pages 995–1072. MIT Press, 1990.

[12] F. Fages. A new fixpoint semantics for general logic programs compared with the wellfounded and the stable model semantics. *New Generation Computing*, 9(4), 1991.

[13] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[14] K. Fine. Some connections between elementary and modal logic. In *Proceedings of the Third Scandinavian Logic Symposium*, pages 15–31, 1975.

[15] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *International Conference on Logic Programming (ICLP 88)*, pages 1070–1080, 1988.

[16] J. Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, University of Amsterdam, 1999.

[17] C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

[18] F. Giunchiglia. Contextual reasoning. *Epistemologia*, XVI:345–364, 1993.

[19] F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics, or: how we can do without modal logics. *Artificial Intelligence*, 65(1):29–70, 1994.

[20] J. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, 1990.

[21] J. McCarthy. Notes on formalizing context. In *International Joint Conference on Artificial Intelligence (IJCAI 93)*, pages 555–560, 1993.

[22] J. McCarthy and S. Buvač. Formalizing context (expanded notes). In *Computing Natural Language*, volume 81 of *CSLI Lecture Notes*, pages 13–50. 1998.

[23] J.-J.Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. Campbridge University Press, 1995.

[24] J. Miller and L. Moss. The undecidability of iterated modal relativization. *Studia Logica*, 79:373–407, 2005.

[25] E. Pacuit and R. Parikh. The logic of communication graphs. In *Declarative Agent Languages and Technologies (DALT 04)*, 2004.

[26] R. Parikh and R. Ramanujam. A knowledge based semantics of messages. *Journal of Language, Logic and Information*, 12(4):453–467, 2003.

[27] M. Pauly. *Logic for Social Software*. PhD thesis, ILLC, University of Amsterdam, 2001.

[28] F. Roelofsen. Exploring logical perspectives on distributed information and its dynamics. *Master Thesis, University of Amsterdam*, 2005.

[29] F. Roelofsen. Non-deterministic multi-context systems. In *European Summer School in Logic, Language, and Information (ESSLLI 05)*, 2005.

[30] F. Roelofsen and L. Serafini. Minimal and absent information in contexts. In *International Joint Conference on Artificial Intelligence (IJCAI 05)*, 2005.

[31] F. Roelofsen, L. Serafini, and A. Cimatti. Many hands make light work: Localized satisfiability for multi-context systems. In *European Conference on Artificial Intelligence (ECAI 04)*, pages 58–62, 2004.

[32] F. Roelofsen and Y. Wang. Contraction semantics for distributed knowledge. In *Belief Revision and Dynamic Logic Workshop at ESSLLI 05*, 2005.

[33] T. Sadzik. Exploring the iterated update universe. *Manuscript, Stanford University*, 2005.

[34] C. Sakama and K. Inoue. An alternative approach to the semantics of disjunctive programs and deductive databases. *Journal of Automated Reasoning*, 13:145–172, 1994.

[35] L. Serafini and P. Bouquet. Comparing formal theories of context in AI. *Artificial Intelligence*, 155:41–67, 2004.

[36] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[37] J. van Benthem. *Modal Correspondence Theory*. PhD thesis, University of Amsterdam, 1976.

[38] J. van Benthem. Extensive games as process models. *Journal of Logic, Language, and Information*, 11:289–313, 2002.

[39] J. van Benthem. One is a lonely number: on the logic of communication. *Technical Report PP-2002-27, ILLC Amsterdam*, 2002.

[40] J. van Benthem. Open problems in logical dynamics. *Technical Report PP-2005-06, ILLC Amsterdam*, 2005.

[41] J. van Benthem. Two logical concepts of information. *Memorial Volume for Jon Barwise*, 2005.

[42] J. van Benthem, J. van Eijck, and B. Kooi. Logics of communication and change. *Technical Report PP-2005-09, ILLC Amsterdam*, 2005.

[43] W. van der Hoek, B. van Linder, and J.-J. Meyer. Group knowledge is not always distributed (neither is it always implicit). *Mathematical social sciences*, 38:215–240, 1999.

[44] H. van Ditmarsch. *Knowledge Games*. PhD thesis, University of Amsterdam & University of Groningen, 2000.

[45] H. van Ditmarsch. The russian cards problem. *Studia Logica*, 75:31–62, 2003.

[46] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer-Kluwer, 2006.

[47] A. van Gelder, K. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.