Logics of Communication and Knowledge

Floor Sietsma

Logics of Communication and Knowledge

ILLC Dissertation Series DS-2012-11

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 904
1098 XH Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

Logics of Communication and Knowledge

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof.dr. D.C. van den Boom
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Agnietenkapel
op donderdag 13 december 2012, te 12.00 uur

door

Floor Anna Gineke Sietsma

geboren te Amstelveen.

Promotiecommissie

Promotoren:	Prof. Dr. J. van Eijck Prof. Dr. K. R. Apt
Overige leden:	Dr. A. Baltag Prof. Dr. J. van Benthem Dr. H. van Ditmarsch Prof. Dr. Y. Venema Prof. Dr. R. Verbrugge

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The work presented in this thesis was carried out at Centrum voor Wiskunde en Informatica (CWI) under the auspices of the Institute for Logic, Language and Computation (ILLC). This research was supported by the Netherlands Organization for Scientific Research (NWO).

Printed and bound by Ipskamp Drukkers.
Cover design based on a stamp by Crafty Individuals.
ISBN: 978-94-6191-503-0

Contents

Acknowledgments

During the course of my PhD project I have grown a lot, both as a person and as an academic. I greatly appreciate the freedom I experienced and the insights I gained, on the subject of logic and on the subject of life. This would not have been possible without the help of many people, and I would like to take this opportunity to thank them.

First of all, I would like to thank my supervisors Jan and Krzysztof.

Jan, you introduced me to logic in the first place and your enthusiasm has greatly inspired me. However dull and tired I felt, after a conversation with you I was always full of energy and motivation again! Moreover, you were ready to help me with whatever issue I had: whether it was a technical question or I needed career advice, you did your best to stop me from worrying and I always felt you had my best interests at heart.

Krzysztof, you were a stable force through my whole PhD project, preserving the structure and making sure that everything was accomplished in the right manner on the right moment. It was a reassuring thought that you were there, with a watchful eye, in case I forgot something. You have an amazing eye for detail, and I cannot even imagine how many mistakes are not in my work because you pointed them out to me!

When I first came to CWI, I got an appointment even though the project I was supposed to work on would not be approved until half a year later. Many thanks to Monique and Paul for making this possible. Also many thanks to NWO for giving me a special personal grant to carry out this research, and to Jan Karel Lenstra for financing an appointment through CWI in case the NWO grant did not work out. In the end this appointment was filled by Sunil Simon.

Sunil, thank you for being such a friendly and social officemate. I feel very comfortable working next to you, and I hope to do so still for some time!

When I started my PhD, Yanjing Wang was my roommate and predecessor as a PhD candidate. Yanjing, thank you for introducing me to the art of being a PhD student. Your amazing work ethics were very inspiring and you were always

ready to help me with whatever problem I had. Also, you showed me what real Chinese food tastes like!

Thanks to Alexandru, Hans, Johan, Rineke and Yde for reading my thesis in a fairly short amount of time. I'm really glad I could defend my thesis so soon. Also many thanks for the detailed comments some of you gave me.

I would like to thank Monique, Paul and Jurgen for being such friendly group leaders and giving me the freedom to work in my own time and style. I always found CWI a very relaxed and friendly environment. Jurgen, your energy and liveliness have really inspired me. Especially our talk during last summer has helped me to write this thesis in a really short time.

Thanks to all my colleagues at SEN1, especially for the funny stories at lunch. Again and again I wonder why I am nerd of the year, instead of you guys. And you may take that as a compliment.

Thanks to all my colleagues at PNA1. I really enjoyed the atmosphere in PNA1, especially at the dinners and outings. Many thanks to Suzanne for organising them!

Thanks to Jan Rutten for taking me to Cyprus! It was my first conference ever and I really enjoyed it, even though I didn't understand most of the talks.

Thanks to Rohit Parikh for inviting me to New York. It was a great experience to be there, and it helped me a lot to focus on my research. Thanks to Joe Halpern for inviting me to Cornell. It was really inspiring to meet you. Also, what a beautiful university! Thanks to Dexter and Alexandra for taking me out to dinner there, it was a memorable night and just what I needed, being alone in a strange city.

During the second year of my PhD, I really enjoyed organising activities for my fellow PhD students. Thanks to the whole committee for doing this with me - we were a good team!

Charlotte, dankjewel dat je zo'n goede vriendin voor me bent. Ik vind het supergezellig dat we nu samen wonen. Nu dit af is heb ik voortaan meer tijd voor gezellig avondjes in de keuken!

Bram, je bent liefste vriend bent die ik me voor kan stellen. Je begrijpt me helemaal, en dat is een klein wonder. Dankjewel dat ik bij je terecht kan als ik ergens mee zit, en dat ook kon tijdens mijn PhD tijd.

Lieve Fee, ik ben zo blij dat jij mijn zusje bent. Wat er ook aan de hand is, jij wil me altijd wel opvrolijken met een knuffel of een kus. Dankjewel voor alle liefde die je me geeft!

Mama, je bent de beste moeder die er bestaat. Jij helpt mij op alle fronten, altijd. Ook aan dit proefschrift heb je op talloze manieren bijgedragen. Dankjewel voor je onvoorwaardelijke steun en liefde. En dankjewel dat je me hebt geleerd wat zonnedoelen zijn.

Chapter 1

Introduction

1.1 Motivation

Communication is all around us. We all communicate as soon as we are among others, which is usually the greatest part of our waking time. We communicate with our friends about the things we did last weekend or the movie we want to see next week. We communicate with our family about who will do the dishes or where we want to go for holidays. And when we go to work, we communicate with our colleagues in order to do our job.

Communication is a very important way for us to influence and interact with the things and people around us. If we were unable to communicate this would entirely change the way we behave and interact. In modern society, communication with our peers has become even more important than the ability to build something ourselves.

We can distinguish many different kinds of communication. There is one distinction that is particularly relevant here. On the one hand there is the live conversation which is a rapid exchange of short messages, usually single sentences. On the other hand there is communication with messages that are sent and received at separate times. These messages are usually longer. Often the first type of communication is spoken and the second is written, but there are exceptions to this rule. For example, instant messaging is a written form of communication of the first type, and recording messages on a voice mail machine is a spoken form of communication of the second type.

Communication and knowledge are closely related. Indeed, the goal of communication is to share information with other people. If this information is known to be truthful, we may call it knowledge. In any case, every successful act of communication creates the knowledge that a certain message is communicated.

Communication can be very simple. For example, when I call my flatmate and tell her that I did the groceries, she will know she will not need to pass by the supermarket on her way home. But there is more to be observed in this

situation: I also know that she knows I did the groceries. This can be quite important because when I realize later on that I forgot something, I will call her again to make sure she *does* pass by the supermarket. Furthermore, she knows that I know that she knows that I did the groceries. Therefore, if she later on realizes she needs something special from the supermarket, she might call me to say that she will be late for dinner because she is going to pass by the supermarket after all. This already shows that even in very simple acts of communication, there is a lot to be analyzed.

But there are also more complex forms of communication. A well known example of this is the Two Generals Problem, first published in [Akkoyunlu et al., 1975] and described in the following form in [Gray, 1978]. Suppose there are two generals, whose armies are situated on opposite hills. In the valley between them is their common enemy and they want to attack this enemy. If one of them attacks on his own he will certainly lose. On the other hand, if they attack together they will probably win. Therefore, they need to coordinate their actions to agree on a common date and time of attack.

They start communicating by sending each other messages. Each messenger will have to pass through the valley where the enemy is encamped, and risks his life by doing so. Therefore, the generals can never be sure that the messages they send out will reach the other hill. Luckily, the generals have their own personal seals which make it impossible for the enemy to fake a message and create false belief among the generals. Will the generals be able to coordinate their attacks?

It may come as a surprise that the answer to this question is "no". To see why this is the case, suppose that the first general sends the following message: "I will attack on Friday morning at nine o'clock!". Now of course this message may not reach the other general, but let us give the generals the benefit of the doubt and suppose the message does reach its destination. Then the second general will know the date and time of attack. But on Friday morning, the first general will discuss with his officers and reason as follows: it could be that his message reached the second general and the second general knows he is supposed to attack today. But it could also be that the messenger was shot on his way, and then the second general will not attack today. Then if I attack now, I will be alone and I will certainly lose. That is a risk the first general is not willing to take.

Therefore the first general changes his message a bit. Instead of just sending the date and time of attack, he also asks the second general to send a messenger back in order to confirm their agreement. Supposing this first message reaches the second general, he will send a messenger back. Suppose this second messenger reaches the first general again. Then on Friday morning, the second general will reason as follows: "If the first general received my confirmation, he will attack with me. But if he did not receive it he will probably not attack and then I will lose!". Therefore, the second general will not attack.

The second general could extend the communication protocol even further by asking the first general to send a confirmation of the confirmation he received,

but this will only move the problem back to the first general, who would then not know whether the second general received his confirmation of the confirmation. This problem cannot be solved: whatever messages the generals send, the one who sent the last message will never know if the other one received it. Therefore they cannot coordinate their attacks without risking to attack on their own.

What the generals lack is exactly the type of knowledge that I shared with my flatmate in the first example. In that situation, she knows I did the groceries, I know she knows this, she knows I know she knows, I know she knows I know she knows it, etcetera ad infinitum. We call this kind of knowledge *common knowledge*: my flatmate and I have common knowledge of the fact that I did the groceries. In the example with the generals, after the first message the second general knows the date and time of attack. After the first general receives a confirmation from him, he will know that the second general knows the date and time. If he also sends a confirmation back and this confirmation reaches the second general, the second general will know that the first general knows that the second general knows the date and time of attack. However, the first general does not know this because he does not know whether the last confirmation reached the second general. In other words, the generals cannot coordinate because they do not share common knowledge.

The difference between the two situations is that when I talk to my flatmate on the phone, I am sure she can hear me. We instantly acquire common knowledge of the content of our conversation. On the other hand, the generals are never sure their message reaches the other side and therefore they cannot create common knowledge. This is an example of unreliable communication. In this work we will mostly assume that messages that are sent by one party are also received by the other party, and that this fact is common knowledge. An exception to this rule is Chapter 6 where we will distinguish between *potential* and *definitive* knowledge. In the example of the generals, the generals have potential knowledge of a message if it is sent to them and they have definitive knowledge of it if they also sent a confirmation of receiving it. Common knowledge is a very important concept which is extensively discussed in this work, especially in Chapter 5.

A fairly new form of communication is email communication. For example, instead of calling my flatmate to tell her I did the groceries, I could send her an email with this information. Email communication can also be more complex: I could include more people as Carbon Copy (CC) recipients in order to start a group conversation over email, or I could even include some Blind Carbon Copy (BCC) recipients who would receive the email without the other recipients being aware of this. In the first case, upon reading the email my flatmate would know that I did the groceries. In the second case, she would also know that the CC-recipients also know this, if they received the message. In the third case, her knowledge would be the same as in the second case because she cannot see the fact that there were BCC-recipients. However, if she takes the time to reflect on all possibilities she might realise that it is possible I included some BCC-

recipients. So in all cases she will consider it possible that more people than that she is aware of received my email.

All these considerations depend greatly on whether we assume that other people read their email. This may be a very reasonable assumption. For example, there are companies where it is required of the employees to check their email daily and read everything of importance. In private communication this is less strict, but even then there are people who can be counted upon to read their email at least daily and sometimes even hourly. On the other hand, there are also people who forget to check their email, or simply do not read all emails they receive. And even if we read our email thoroughly, there are spam filters that may accidentally remove email from our inbox or network errors that may result in emails being lost.

In the example with me and my flatmate, the situation can be analyzed easily by hand. It is not hard to figure out who knows what by just looking at the email I sent her. However, when the number of emails and recipients grows this analysis becomes a daunting task and infeasible for humans. For example, in large companies tens of thousands of emails are sent and received every day. When some secret piece of information was leaked via email, the complexity of finding out who was the source of this information leak, or who else received this secret information, is overwhelming. In situations like this, it would be very helpful if the analysis of people's knowledge during an email conversation could be automated. Due to the intricacies involved when studying knowledge, knowledge about knowledge and common knowledge, logic is a very suitable tool for such an analysis. This explains the extensive reliance on logic in this thesis.

1.2 Overview of the dissertation

The general set-up of this dissertation is as follows. I first give some preliminary definitions in Chapter 2. In Chapters 3, 4, 5 and 6, I present four different models of how knowledge evolves during communication. Each of these models depends on different assumptions and is therefore suitable for different situations. Chapter 3 focuses on a situation where all possible messages are known by the agents, for example during a game or during the execution of some protocol. The model presented in Chapter 4 is a very general model which can be used to model many types of communication. It is not tailored towards one single situation, but can be adapted as desired. Chapter 5 and 6 focus specifically on email communication. The model presented in Chapter 5 is of a more theoretical nature and focuses on modeling common knowledge. Also, it rests on the assumption that all emails that are sent are also received and read. On the other hand, the model presented in Chapter 6 distinguishes two kinds of knowledge in order to make a distinction between when an email is sent and when it is also read. In Chapter 7, I take a closer look at the models that are used in Chapter 3. These are the so-called

action models that are used in epistemic logic to model communicative actions. In Chapter 8 I show how these models can be used in communication about beliefs and belief revision and finally, in Chapter 9 I study a situation in which the agents that communicate are not necessarily truthful, which leads to a study of the effect of lying. I also present a case study of a game of Liar's Dice.

The contents of each chapter is briefly sketched below.

Chapter 2 This is an introductory chapter explaining some basic concepts from epistemic logic.

Chapter 3 In this chapter I propose a framework for modeling message passing situations that combines the best properties of dynamic epistemic semantics and history-based approaches. I assume that all communication is truthful and reliable. I also assume there is a dynamic set of messages that may be sent, which is known by all agents. The framework consists of Kripke models with records of sent messages in their valuations. I introduce an update operation for message sending. With this update I can study the exact epistemic consequences of sending a message. I define a class of models that is generated from initial Kripke models by means of message updates, and axiomatize a logic for this class of models. Next, I add an update modality and sketch a procedure for defining it by means of equivalence axioms. This chapter is based on joint work with Jan van Eijck [Sietsma and van Eijck, 2011].

Chapter 4 In this chapter, I develop a very general framework based on epistemic logic that can be adapted to the needs of a great number of different situations. The network over which the agents communicate is explicitly specified in this framework, and therefore it can be used to model a situation where not all agents are able to communicate with each other. By combining ideas from Dynamic Epistemic Logic and Interpreted Systems, the semantics offers a natural and neat way of modeling multi-agent communication scenarios with different assumptions about the observational power of agents. I relate the logic to the standard DEL and IS approaches and demonstrate its use by studying a telephone call communication scenario. This chapter is based on joint work with Yanjing Wang and Jan van Eijck [Wang et al., 2010].

Chapter 5 Here, I focus on email communication specifically. I consider a framework in which a group of agents communicates by means of emails, with the possibility of replies, forwards and BCC. I study the epistemic consequences of such email exchanges by introducing an appropriate epistemic language and semantics. This allows me to find out what agents exactly learn from the emails they receive. Common knowledge plays a big role in this framework and I show how to determine when a group of agents

acquires common knowledge of the fact that an email was sent. I also give an analysis of BCC and I look at email communication from the perspective of distributed systems. This chapter is based on joint work with Krzysztof Apt [Sietsma and Apt, 2012].

Chapter 6 In this chapter I also analyze email communication, but now I focus on the difference between sending an email and knowing its content has been read. This is not the same thing, especially when one considers the existence of network errors, spam filters and people who simply do not read all the emails they receive. Such an analysis is interesting in many situations. One example is when someone's knowledge about some email at a particular moment may be relevant in a court case. I distinguish two kinds of knowledge: potential knowledge, which is acquired at the moment an email is sent to someone, and definitive knowledge, which is acquired when that person also shows his knowledge of the email by replying to it or forwarding it. I incorporate both kinds of knowledge in my logic. I present a semantics for this logic that can be decided quite easily and is therefore applicable in practice. I also show that from the epistemic point of view, the BCC feature of email systems cannot be simulated using messages without BCC recipients. This chapter is based on an unpublished manuscript that I finished in 2012.

Chapter 7 In this chapter I take a closer look at the models I use in Chapters 3 and 9. These are Kripke models, used to model knowledge in a static situation, and action models, used to model communicative actions that change this knowledge. The appropriate notion for structural equivalence between modal structures such as Kripke models is bisimulation: Kripke models that are bisimilar are modally equivalent. I would like to find a structural relation that can play the same role for the action models that are of great importance in information updating. Two action models are equivalent if they yield the same results when updating Kripke models. More precisely, two action models are equivalent if it holds for all Kripke models that the result of updating with one action model is bisimilar to the result of updating with the other action model. In this chapter I propose a notion of action emulation that characterizes the structural equivalence of the important class of canonical action models. Since every action model has an equivalent canonical action model, this gives a method to decide the equivalence of any pair of action models. I also give a partial result that holds for the class of all action models. This chapter is based on joint work with Jan van Eijck [Sietsma and van Eijck, 2012].

Chapter 8 This chapter focuses on the interplay between knowledge and belief. Models of knowledge change into models of belief when one drops the

assumption that all communication is truthful. This corresponds to the assumptions that all relations in the Kripke models are equivalence relations. In this chapter, the only constraint I impose on these relations is that they are linked. Linkedness is a new extension of the notion of local connectedness for multiple agents. It assures that if there are three alternatives, one agent prefers the second over the first, and the other agent the third over the first, that both agents make up their mind about whether they prefer the second or the third alternative. This is important in consensus-seeking procedures like Dutch meetings, where the participants vote on different subjects according to a set agenda. I show how my framework can be used to model such procedures, and use it to analyze the discursive dilemma, a well known problem in judgement aggregation [List and Pettit, 2005]. This chapter is based on joint work with Jan van Eijck [Sietsma and van Eijck, 2008].

Chapter 9 This chapter has a more philosophical flavor as compared to the other, more technical, chapters. I model lying as a communicative act changing the beliefs of the agents in a multi-agent system. Following St. Augustine, I see lying as an utterance believed to be false by the speaker and uttered with the intent to deceive the addressee. The deceit is successful if the lie is believed by the addressee. I provide a logical sketch of what goes on when a lie is communicated. I present a complete logic of manipulative updating, to analyze the effects of lying in public discourse. Next, I turn to the study of lying in games, in particular the game of Liar's Dice. First, a game-theoretical analysis explains how the possibility of lying makes such games interesting, and how lying is put to use in optimal strategies for playing the game. I also give a matching logical analysis for the games perspective, and implement that in the model checker DEMO. There is a difference between lying in games and the logical manipulative update: instead of taking each utterance to be truthful, in a game the players are aware of the fact that the other players may lie. This chapter is based on joint work with Hans van Ditmarsch, Jan van Eijck and Yanjing Wang [van Ditmarsch et al., 2012].

Chapter 2

Preliminaries

In this chapter I will explain some preliminaries that are useful for understanding the other chapters of the thesis. I will introduce Kripke models, which can be used to represent the knowledge of agents in some static situation. I will also discuss action models, which can be used to update Kripke models when the situation changes. I will use these models later on to reason about the knowledge of agents during some message exchange, using Dynamic Epistemic Logic.

2.1 Dynamic Epistemic Logic

2.1.1. DEFINITION. Let a set of agents Ag and a set of propositions P be given. A ***Kripke model*** for Ag and P is a tuple $\mathcal{M} = (W, R, Val, W_0)$ where W is a set of worlds, R is a function that assigns to each $a \in Ag$ an equivalence relation R_a on W, Val is a function that assigns to each world in W a subset of P (its valuation), and $W_0 \subseteq W$ is the set of actual worlds. I will sometimes use $\sim_A$ for R_a. Given a Kripke model $\mathcal{M}$, I use $W^{\mathcal{M}}, R^{\mathcal{M}}, Val^{\mathcal{M}}, W_0^{\mathcal{M}}$ to denote its elements.

The interpretation of these Kripke models is as follows. The worlds in W are different scenarios the agents consider possible. In each world each proposition has a truth value given by the valuation of that world. There is a relation between two worlds w_1 and w_2 for an agent a if, when in situation w_1, agent a considers it possible that instead of w_1, w_2 is the case. In other words, agent a does not have the knowledge to distinguish situation w_1 from situation w_2. The worlds in W_0 are the actual worlds, the situations that are considered possible by the designer of the model.

To describe and reason about the exact knowledge of the agents I will use epistemic Propositional Dynamic Logic (PDL) [Kozen and Parikh, 1981].

2.1.2. DEFINITION. Given some set of propositions P and a set of agents Ag, let

$\mathcal{L}$ be the language consisting of formulas of the form ϕ as given below.

$$\begin{array}{lll} \phi & ::= & p \mid \neg\phi \mid \phi \vee \phi \mid \langle\alpha\rangle\phi \qquad \text{where } p \in P, \\ \alpha & ::= & a \mid ?\phi \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha^* \qquad \text{where } a \in Ag. \end{array}$$

Call α an ***epistemic program***.

I use the usual abbreviations: $\phi \wedge \psi$ for $\neg(\neg\phi \vee \neg\psi)$ and $[\alpha]\phi$ for $\neg\langle\alpha\rangle\neg\phi$.

This language can be interpreted on the worlds of a Kripke model. The epistemic programs α represent relations that are built from the knowledge relations of the agents. The program a stands for the relation of agent a. The program $?\phi$ goes from any world in the Kripke model to itself, if and only if that that world satisfies ϕ. It can be used to test the truth value of ϕ. The program $\alpha_1;\alpha_2$ is the sequential composition of α_1 and α_2: it goes from one world to another if there is an α_1 relation from the first world to a third world, and an α_2 relation from the third world to the second world. The program $\alpha_1 \cup \alpha_2$ is the choice between α_1 and α_2: it goes from one world to another if there is either an α_1 or an α_2 relation between them. Finally, the α^* relation stands for repeating α finitely many times: it goes from one world to another if the second world can be reached from the first one by following a finite number of α relations.

The formula $\langle\alpha\rangle\phi$ holds in a world if there is an α-related world that satisfies ϕ. Dually, $[\alpha]\phi$ holds if all α-related worlds satisfy ϕ. Given some agent a, $\langle a\rangle\phi$ holds if a thinks it possible that ϕ. On the other hand, $[a]\phi$ holds if a knows that ϕ is true.

The formal definition of the semantics is given below. Given some program α, $[\![\alpha]\!]^{\mathcal{M}}$ denotes the relation that interprets the program α in $\mathcal{M}$.

2.1.3. Definition. Let $\mathcal{M} = (W, R, Val, W_0)$ be a Kripke model. Then the truth of an $\mathcal{L}$ formula ϕ is given by:

$$\begin{array}{lll} \mathcal{M} \models_w p & \text{iff} & p \in Val(w) \\ \mathcal{M} \models_w \neg\phi & \text{iff} & \mathcal{M} \not\models_w \phi \\ \mathcal{M} \models_w \phi_1 \vee \phi_2 & \text{iff} & \mathcal{M} \models_w \phi_1 \text{ or } \mathcal{M} \models_w \phi_2 \\ \mathcal{M} \models_w \langle\alpha\rangle\phi & \text{iff} & \exists w' : w[\![\alpha]\!]^{\mathcal{M}} w' \text{ and } \mathcal{M} \models_{w'} \phi \end{array}$$

$$\begin{array}{lll} w[\![a]\!]^{\mathcal{M}} w' & \text{iff} & w \sim_a w' \\ w[\![?\phi]\!]^{\mathcal{M}} w' & \text{iff} & w = w' \text{ and } \mathcal{M} \models_w \phi \\ w[\![\alpha_1;\alpha_2]\!]^{\mathcal{M}} w' & \text{iff} & \exists w'' \in W : w[\![\alpha_1]\!]^{\mathcal{M}} w'' \text{ and } w''[\![\alpha_2]\!]^{\mathcal{M}} w' \\ w[\![\alpha_1 \cup \alpha_2]\!]^{\mathcal{M}} w' & \text{iff} & w[\![\alpha_1]\!]^{\mathcal{M}} w' \text{ or } w[\![\alpha_2]\!]^{\mathcal{M}} w' \\ w[\![\alpha^*]\!]^{\mathcal{M}} w' & \text{iff} & \exists w_1, \ldots, w_n \in W : w_1 = w, w_n = w' \text{ and} \\ & & w_1[\![\alpha]\!]^{\mathcal{M}} w_2[\![\alpha]\!]^{\mathcal{M}} \ldots [\![\alpha]\!]^{\mathcal{M}} w_n. \end{array}$$

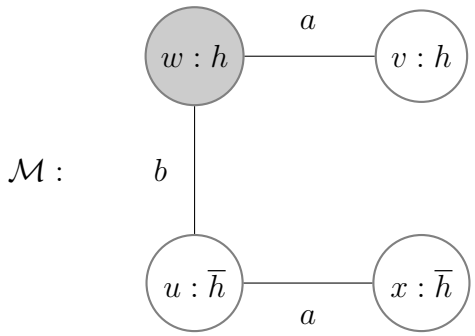
In the last part of this definition, note that $w[\![\alpha^*]\!]w'$ if and only if there is a path from w to w', which holds in particular when $w = w'$.

The relations in the Kripke models are often constrained in order to impose restrictions on the knowledge of the agents. For example, true knowledge is represented by Kripke models with relations that are reflexive, symmetric and transitive. Reflexivity means that there is a relation from every world to itself. It corresponds to the axiom $[a]\phi \rightarrow \phi$, which expresses that if an agent knows something, then it is true. Symmetry means that if there is a relation from world w to world v, then there is also a relation back from v to w. It is characterized by the axiom $\phi \rightarrow [a]\langle a\rangle\phi$, which expresses that if ϕ is true then every agent knows that it is possible that ϕ is true. Transitivity means that if there is a relation from w to v, and from v to u, then there is also a relation from w to u. In other words, if there is a path from one world to a second one through other worlds, then there is also a direct relation. It is characterized by the axiom $[a]\phi \rightarrow [a][a]\phi$, which expresses that if an agent knows something then she knows that she knows it. Relations that are reflexive, symmetric and transitive are called equivalence relations, and Kripke models of which all relations are equivalence relations are called S5 models. They are used to model knowledge. Another class of models I will use is the class of KD45 models, that are used to model belief instead of knowledge. They have relations that are transitive, serial and euclidean. Seriality means that for every w, there is a relation to some world v. Euclideanness means that for every w, v and u such that there is a relation from w to v and one from w to u then there is also one from v to u. In Chapers 3, 4, 5 and 6 I will work with epistemic relations that are equivalence relations. Chapter 7 does not assume any restrictions on the relations, and Chapter 8 will propose a new restriction, namely linkedness. Finally, in Chapter 9 I will focus on KD45 models.

Here, I will first show how Kripke models can be used with a clarifying example.

2.1.4. Example. Suppose there are two people, Alice and Bob, who are playing a game together. They flip a coin under a cup, in such a way that the result is hidden. Then, Alice looks under the cup and sees that the coin is heads. Now Alice leaves the room to go to the toilet. When she comes back, she does not know whether Bob has secretly looked under the cup, so she does not know whether Bob knows it is heads. Actually, Bob is a very honest person and he has not looked.

The model for this situation looks as follows:



Here w, v, u and x are the names of the four worlds. The result of the coin flip is represented by the proposition h, where h denotes that h is true and the coin lies heads up and $\overline{h}$ denotes that h is false and the coin lies tails up. The gray colour of the world w denotes that it is an actual world. In this picture I have omitted the reflexive relations, which are present for every agent from every world to itself. Furthermore, since all relations are symmetric I use lines instead of arrows to represent them. I will continue this convention for S5 models in the remainder of this dissertation.

In the actual world w, the coin lies heads up. Alice knows this: the only other world she cannot distinguish from w is world v, where the coin is also heads up. So h holds in every a-related world, and $\mathcal{M} \models_w [a]h$. Bob does not know that the coin lies heads up: there is a relation from w to u, where h does not hold. So $\mathcal{M} \models_w \neg[b]h$.

Now look at v instead of w. There, there is no other world that Bob cannot distinguish from v, so Bob knows that the coin lies heads up: $\mathcal{M} \models_v [b]h$. Since Alice confuses the actual world w with the world v, Alice considers this situation possible. So $\mathcal{M} \models_w \langle a\rangle[b]h$: in the actual world w, Alice holds it possible that Bob knows h. This follows from the semantics because there is an a-relation from w to v, and no b relation from v to a world where h does not hold.

Bob does not know the result of the coin flip. Bob does know that Alice holds it possible that Bob has looked under the cup. So Bob confuses the actual world where h is true and Alice holds this possible with a world where h is false and Alice holds this possible. This is world u in the model. Because there is a relation for Alice to world x, and in world x the formula $[b]\neg h$ holds, the world u satisfies $\langle a\rangle[b]\neg h$: Alice holds it possible that Bob knows $\overline{h}$. Because there is a relation from w to u, Bob thinks this formula might be true: in the actual world, $\langle b\rangle\langle a\rangle[b]\neg h$ holds. Intuitively, Bob considers it possible that h is false and that Alice thinks Bob might know this.

Using epistemic programs, more complex notions of knowledge can be expressed. For example, one could say that Alice thinks it is possible that Bob thinks it is possible that h is not true with the formula $\langle a; b\rangle\neg h$. It holds in v because there one can follow an a-relation and then a b-relation to a $\neg h$-world u, but also in w because there is a reflexive a-relation from w to itself (not shown

in the picture) that can be followed from w to w, after which a b-relation can be followed to u.

Another property of the model is that in world w both Alice and Bob know that Alice knows the value of h. This can be expressed as $[a \cup b]([a]h \vee [a]\neg h)$. The modality $[a \cup b]$ expresses that both a and b know something. There is even something stronger that holds: it is ***common knowledge*** among Alice and Bob that Alice knows the value of h. This means that they both know it, and both know that the other knows it, and both know the other knows they know it, etcetera. It is expressed by $[(a \cup b)^*]([a]h \vee [a]\neg h)$. In general, given a finite group of agents $a_1, ..., a_n$, $[(a_1 \cup ... \cup a_n)*]$ denotes common knowledge within the group.

Sometimes, two different Kripke models represent exactly the same situation. In this case they are equivalent. Such an equivalence can be detected by checking whether there exists a bisimulation between the models. This is a relation between the worlds of the models that has certain special properties.

2.1.5. Definition. Given two Kripke models $\mathcal{M}$ and $\mathcal{N}$, a relation $Z : W^{\mathcal{M}} \times W^{\mathcal{N}}$ is a ***bisimulation*** if for any $w \in W^{\mathcal{M}}$ and $v \in W^{\mathcal{N}}$ such that $(w, v) \in Z$ the following conditions hold:

Invariance $Val^{\mathcal{M}}(w) = Val^{\mathcal{N}}(v)$,

Zig for any agent $a \in Ag$, if there is a world w' such that $w \sim_a^{\mathcal{M}} w'$ then there must be a world v' such that $v \sim_a^{\mathcal{N}} v'$ and $(w', v') \in Z$,

Zag for any agent $a \in Ag$, if there is a world v' such that $v \sim_a^{\mathcal{N}} v'$ then there must be a world w' such that $w \sim_a^{\mathcal{M}} w'$ and $(w', v') \in Z$.

I write $(\mathcal{M}, w) \leftrightarrows (\mathcal{N}, v)$ if there exists a bisimulation between $\mathcal{M}$ and $\mathcal{N}$ that links $w \in W^{\mathcal{M}}$ and $v \in W^{\mathcal{N}}$. If there exists a total bisimulation between the worlds in $W_0^{\mathcal{M}}$ and $W_0^{\mathcal{N}}$ I write $\mathcal{M} \leftrightarrows \mathcal{N}$ and say that $\mathcal{M}$ and $\mathcal{N}$ are ***bisimilar***.

So two bisimilar worlds satisfy the same propositions, and if one of these worlds has a relation to a third world then the other should have a relation to a fourth world that is bisimilar to the third world.

The following result is standard in modal logic, see for example [Blackburn et al., 2001]:

2.1.6. Theorem. *If* $(\mathcal{M}, w) \leftrightarrows (\mathcal{N}, v)$ *then for any modal formula* φ*,*

$$\mathcal{M} \models_w \varphi \text{ iff } \mathcal{N} \models_v \varphi.$$

All formulas I will consider in this thesis are modal formulas, so for all my purposes bisimilar worlds may be considered equivalent.

Sometimes I will be interested in a bisimulation that takes only certain propositions into account. A ***restricted bisimulation*** for $Q \subseteq P$ is a relation that

satisfies the conditions for bisimulation when taking for the invariance condition only the propositions in Q into account. If two worlds are related by such a relation then they are Q-bisimilar, notation: $(\mathcal{M}, w) \leftrightarrows_Q (\mathcal{N}, v)$. So the truth value of propositions in $P \setminus Q$ may differ between Q-bisimilar worlds.

Kripke models represent the knowledge of agents in a static situation. When communication takes place, the situation changes. Therefore, the Kripke model needs to be changed as well. I use action models, introduced in [Baltag et al., 1998], to represent a communicative event that changes the knowledge of agents. In particular, I use them to represent the event that some message is sent.

An action model is like a Kripke model, only instead of possible worlds it has possible events which have a formula called a precondition instead of a valuation. Action models can be applied to Kripke models in order to update them. Then every world from the Kripke model gets matched with every event from the action model, provided that the world satisfies the precondition of the event. This operation is called the product update.

Formally, an action model is defined as follows:

2.1.7. Definition. Let a set of agents Ag and a set of propositions P be given. An ***action model*** for Ag and P is a tuple $\mathcal{A} = (E, R, Pre, E_0)$ where E is a set of events, R is a function that assigns to each $a \in Ag$ an equivalence relation R_a on E, Pre is a function that assigns to each event in E an $\mathcal{L}$-formula over P (its precondition), and $E_0 \subseteq E$ is the set of actual events. I will sometimes use $\sim_a$ for R_a, and I will use $E^{\mathcal{A}}, R^{\mathcal{A}}, Pre^{\mathcal{A}}, E_0^A$ to denote the elements of the action model.

When a Kripke model is updated with an action model, the knowledge of the agents represented in the model is changed by changing the relations between the worlds. If there is a relation between two worlds in the Kripke model and these worlds are matched with two events in the action model, then the relation is only preserved if there is also a relation between the two events in the action model.

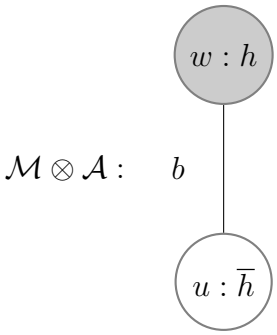
The formal definition of the product update is as follows:

2.1.8. Definition. Given a Kripke model $\mathcal{M}$ and an action model $\mathcal{A}$, the result of updating $\mathcal{M}$ with $\mathcal{A}$ is the model $\mathcal{M} \otimes \mathcal{A} = (W', R', Val', W_0')$ given by

$$\begin{array}{rcl} W' & := & \{(w, e) \mid w \in W^{\mathcal{M}}, e \in E^{\mathcal{A}}, \mathcal{M} \models_w Pre^{\mathcal{A}}(e)\}, \\ (w, d)R'_a(v, e) & \text{iff} & wR_a^{\mathcal{M}}v \text{ and } dR_a^{\mathcal{A}}e, \\ Val'((w, e)) & := & Val^{\mathcal{M}}(w), \\ W_0' & := & \{(w, e) \in W' \mid w \in W_0^{\mathcal{M}} \text{ and } e \in E_0^{\mathcal{A}}\} \end{array}$$

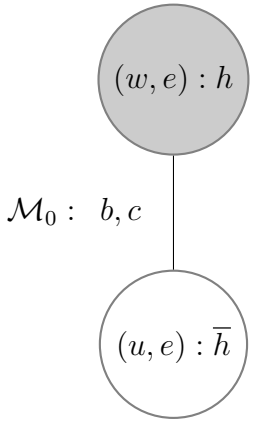
2.1.9. Example. Consider the situation from the previous example. If someone would come into the room and announce that Bob has not looked under the cup, then the knowledge of Alice would change. She would get to know that Bob does not know the result of the coin flip. The action model for this looks as follows:

It has one world with precondition $\neg[b]h$. The result of this action is that only the worlds in the Kripke model that satisfy this precondition, are preserved in the result of the update. When I update the Kripke model from Example 2.1.4 with this action model, I get the following result:

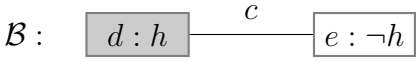


Here, world v has been removed because it did not satisfy $\neg[b]h$. Now, in the actual world w, Alice knows that Bob does not know the result of the coin flip: $\mathcal{M} \otimes \mathcal{A} \models_w [a]\neg[b]h$.

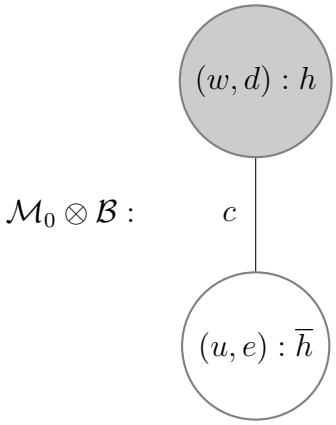
This is a quite simple action model: it has only one world. In order to show an example of a more complex action model, let me introduce another agent, Carol, who does not know the result of the coin flip. I take a new Kripke model for this situation:



In this situation, Alice knows h, Bob and Carol do not, and everyone is aware of each other's knowledge. Suppose now that Alice tells Bob the result of the coin flip. Carol is aware of the fact that Alice tells Bob the truth value of h, but she does not get to know what that value is. The action model for this looks as follows.



There are two possible events: one where Alice tells Bob the coin lies heads up, and one where she tells him it lies tails up. Carol is the only agent who does not know which of the two events is happening, so she confuses the two worlds. Actually, Alice tells Bob the result of the coin flip was heads. When I update the Kripke model with this action model the result is as follows:



Because h is true in w, this world matches with the event d. Because h is false in u, u matches with e. Because there is no b-relation between d and e, the b-relation between w and u is not preserved. This is exactly what is required because now Bob knows the result of the coin flip, so he can distinguish the two situations.

Chapter 3

Message Passing in Dynamic Epistemic Logic

3.1 Introduction

In this chapter I show how one can model the dynamics of knowledge during communication using epistemic logic. I will focus on a situation where a number of agents communicate using messages from a finite set which is known by all agents. This set is not fixed: during the message exchange, new messages may be added to the set. Such a set up is relevant in numerous situations. For example, one could think of computers communicating in accordance with a fixed protocol, or people playing a game where they have to give certain signals every round. The example of the two generals mentioned in the introduction could also be modeled this way, where the possible messages are the possible days of attack. However, in this chapter I will assume that the communication channel is reliable, so every message that is sent is also received. This is clearly not the case for the generals. I also assume that the communication is synchronous, so all the messages that are sent are immediately received.

I will use Kripke models to model the state of knowledge at a particular moment. Given some message that is sent and received, I use its structure to generate the Kripke model that represents the new state of knowledge after reception of the message. This way, sequences of Kripke models can be constructed from sequences of messages. These sequences show how the knowledge of the agents changes over time.

The system is designed for reasoning about sequences of messages that have been sent and received, given some initial situation represented by a Kripke model. The semantics is designed in a back-and-forth fashion: a Kripke model of the current situation determines which communication steps are successful on that model, and each communication step gives rise to an adaptation of the model to a new Kripke model, which again determines which successive communication steps are possible, etcetera.

In this chapter I only consider truthful communication. This means that the content of all the messages that are sent is true. Furthermore, a message can only be sent if the sender of the message knows that its content is true. Also, all messages are accepted as true, so if an agent receives a message she gains knowledge of the contents.

The semantics presented here can be used to model reasoning about the way the communication took place: agents remember which messages they sent or received, but are uncertain about which other messages were sent. This engenders uncertainties about what other agents know and about what messages they may have exchanged. The construction given in this chapter models these uncertainties in a very precise way.

The semantics allows for checking properties and effects of communication sequences that took place in the past, and allows a limited amount of reasoning about counterfactual situations, like "suppose instead of actual message m, another message, m' had been sent." Also, it allows for reasoning about properties and effects of new communication steps.

In the next section I start out with defining a logical language based on messages with a certain internal structure. In Section 3.3 I show how I use Kripke models to interpret this language and I introduce the update that models the communicative action of sending a message. In Section 3.4 I define a class of Kripke models that are a realistic result of a sequence of messages. In Section 3.5 I axiomatize the language and the two new modalities I have introduced. Finally, in Section 3.6 I discuss some related work and I conclude this chapter in Section 3.7.

3.2 The Language of Knowledge and Messages

In this section I will show how to incorporate messages in the epistemic language introduced in Chapter 2. Including these messages in the language allows for reasoning about how the knowledge of agents is affected by messages and the knowledge of the agents about these messages.

I will first define a simple language $\mathcal{L}_0^{MPD}$ that does not contain any knowledge modalities. I will use this language to represent the semantic content of the messages. Later on, I will define a richer language that can be used to reason about the messages and the knowledge of the agents.

Let P be a set of proposition letters. Let Ag be a finite set of agents.

3.2.1. Definition. Let $\mathcal{L}_0^{MPD}$ be the following language:

$$\begin{aligned}\psi ::= \;& p \mid (a, \psi, G) \mid \neg\psi \mid (\psi \vee \psi) \\ & \text{where } p \in P, a \in G \subseteq Ag.\end{aligned}$$

This is propositional logic enriched with messages. A ***message*** is represented by a tuple (a, ψ, G) where $a \in Ag$ is the sender of the message, $\psi \in \mathcal{L}_0^{MPD}$ is the contents of the message and $G \subseteq Ag$ is the group of recipients of the message. The formula (a, ψ, G) expresses that message (a, ψ, G) was sent at some moment in the past.

I adopt the convention that a sender always receives a copy herself: any message (a, ψ, G) has $a \in G$. I will abbreviate $(a, \psi, \{a, b\})$ (a message with a single recipient, plus a copy to the sender) as (a, ψ, b).

I adopt the usual abbreviations: $\psi_1 \wedge \psi_2$ for $\neg(\neg\psi_1 \vee \neg\psi_2)$ and $\psi_1 \to \psi_2$ for $\neg\psi_1 \vee \psi_2$.

The following is a first example of what these messages look like and how they can mention previous messages.

3.2.2. Example. Reply on a message (a, p, b) with a quotation of the original message and some new information q can be expressed as $(b, q \wedge (a, p, b), a)$. Forwarding of (a, p, b) by agent b to some other agent c can be expressed as $(b, (a, p, b), c)$.

This example already shows that notation can become a bit thick when nesting messages. Therefore I will often shorten notation by naming the messages m, m', m_1, etc. These names should be seen as pure abbreviations. If a message (a, ψ, G) is abbreviated as m then I mean with $s_m = a$ the sender of the message, $c_m = \psi$ the content of the message and $r_m = G$ the group of recipients of the message. I also use these abbreviations in the content of other messages: for example, (b, m, c) is an abbreviation for the message $(b, (a, \psi, G), c)$.

3.2.3. Example. If m is a message, then the message $(a, \neg m, b)$ quotes message m. The *formula* $\neg m$ expresses that m was not sent. With the message $(a, \neg m, b)$, agent a informs agent b that m was not sent. The formula $\neg(a, \neg m, b)$ expresses that the message $(a, \neg m, b)$ was not sent.

Note that the definition of $\mathcal{L}_0^{MPD}$ contains mutual recursion: formulas may contain messages which contain formulas. Due to this mutual recursion the language $\mathcal{L}_0^{MPD}$ is already quite expressive. Even though the content of the messages cannot contain epistemic operators, a considerable number of useful communicative situations can be expressed.

3.2.4. Example. **Send** Communication step consisting of a single message m.

Acknowledgement Acknowledgement of the receipt of a message m can be expressed as (b, m, s_m) where $b \in r_m$.

Reply Reply to sending of m with reply-contents ψ can be expressed as $(b, m \wedge \psi, s_m)$ where $b \in r_m$.

Forward Forwarding of m can be expressed as (b, m, c) where $b \in r_m$ and $c \notin r_m$.

Forward with annotation Forwarding of m with annotation ψ can be expressed as $(b, m \wedge \psi, c)$ where $b \in r_m$ and $c \notin r_m$.

CC There is no distinction between addressee list and CC-list. The distinction between addressee and CC-recipient is in general a subtle matter of etiquette: usually, an addressee is supposed to reply to a message while someone on a CC-list incurs no such obligation. I think it is safe to ignore the difference here.

BCC A message m with BCC recipients $b_1, \ldots, b_n$ can be treated as a sequence of messages m, (s_m, m, b_1), $\ldots$, (s_m, m, b_n). Each member on the bcc list of m gets a separate message from the sender of m to the effect that message m was sent. In Chapter 5 I will discuss a subtle difference between such a "sequence of forwards" and the actual BCC feature. I will prove in Theorem 3.3.8 that the order in which the list (s_m, m, b_1), $\ldots$, (s_m, m, b_n) is sent does not matter.

I will set up the semantics in such a way that I can prove that any message that is forwarded was already sent at some earlier stage, and an acknowledgement never precedes a send. The fact that these properties follow from the epistemic effects of message passing is a corroboration of the appropriateness of my set-up.

The truth value of an $\mathcal{L}_0^{MPD}$ formula depends not only on the truth value of the propositions in P, but also on the truth value of the messages mentioned in the formula. For messages, a positive truth value means that the message was sent, and a negative truth value that it was not sent. In order to know which messages should be considered I first assign a vocabulary to every formula. This is the set of all propositions and messages that are relevant to the truth value of the formula.

3.2.5. Definition. The vocabulary $voc(\varphi)$ of a formula φ is defined as follows:

$$\begin{aligned} voc(p) &:= \{p\} \\ voc((a, \psi, G)) &:= \{(a, \psi, G)\} \cup voc(\psi) \\ voc(\neg\psi) &:= voc(\psi) \\ voc(\psi_1 \vee \psi_2) &:= voc(\psi_1) \cup voc(\psi_2) \end{aligned}$$

The following example shows how this definition works out:

3.2.6. Example. If $m = (a, p \vee q, b)$ and $m' = (b, m, c)$, then

$$voc(m') = \{p, q, m, m'\}.$$

There is an obvious partial order on the vocabulary of a formula. Note that vocabulary elements are either proposition letters or messages. These can be viewed as formulas, which have a vocabulary themselves. Letting x, y range over vocabulary elements, I set $x \preceq y$ if $x \in voc(y)$. I set $x \prec y$ if $x \preceq y$ and $x \neq y$. This partially orders a vocabulary by 'depth of embedding'. For example 3.2.6, this gives $p, q \prec m \prec m'$.

This can be used to define vocabularies *per se*. A ***vocabulary*** is a set of messages and proposition letters that is closed under applications of *voc*. Intuitively, what this means is that if a vocabulary contains m, then it also contains every proposition or message that is mentioned in m.

It is easy to see from this definition that the vocabulary of a formula, and hence also the vocabulary of a finite set of formulas, is always finite. Now I can give a truth definition for formulas of $\mathcal{L}_0^{MPD}$ given some valuation of their vocabulary:

3.2.7. Definition. Let Ψ be a set of $\mathcal{L}_0^{MPD}$ formulas. Let v be a subset of $voc(\Psi)$, representing the propositions that are true and the messages that are sent. Call v a ***valuation*** for Ψ. Then truth at v is defined as follows for all formulas in Ψ:

$$\begin{array}{lll} v \models \top & & \text{always} \\ v \models p & \text{iff} & p \in v \\ v \models m & \text{iff} & m \in v \\ v \models \neg\psi & \text{iff} & v \not\models \psi \\ v \models \psi_1 \wedge \psi_2 & \text{iff} & v \models \psi_1 \text{ and } v \models \psi_2 \end{array}$$

Truth of m at v expresses that according to v message m was sent (at some time in the past).

As mentioned above, I will use a richer language with knowledge modalities to reason about the knowledge of agents and how this is influenced by message passing. I adapt the language from Chapter 2 to include messages, which leads to the following definition of the language $\mathcal{L}^{MPD}$.

$$\begin{array}{lll} \phi & ::= & \psi \mid \neg\phi \mid \phi \vee \phi \mid \langle\alpha\rangle\phi \qquad \text{where } \psi \in \mathcal{L}_0^{MPD} \\ \alpha & ::= & a \mid ?\phi \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha^* \qquad \text{where } a \in Ag. \end{array}$$

The semantics of this language interpreted on the world of a Kripke model is as follows. For the base case $\phi = \psi \in \mathcal{L}_0^{MPD}$ it is given by Definition 3.2.7, with respect to the valuation of the world under consideration. For the other clauses it is as in Chapter 2. Of course, this depends on a vocabulary of propositions and messages. Therefore, I will introduce Kripke models with vocabularies in the next section.

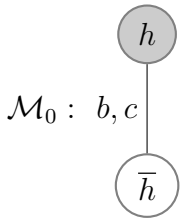
3.3 Modeling Message Passing

I will use Kripke models to represent the knowledge of agents during a sequence of message exchanges. Usual Kripke models only consider the agents' knowledge about basic propositions. Now I also want to consider their knowledge about messages that may have been sent. In order to do this, I will explicitly add messages to the models. Because the size of the models usually increases drastically with the number of messages in the model, I propose the following modeling procedure. Model the initial situation where no messages are sent with a model with no messages. Then gradually add messages to the model as they are sent, and update the models with information concerning who knows about the messages and who does not. The following example illustrates the idea.

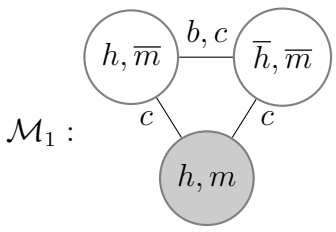
3.3.1. Example. Suppose the initial state is the model $\mathcal{M}_0$ from Example 2.1.9 where Alice knows about h, while Bob and Carol do not. Let m be the message (a, h, b) sent by Alice, informing Bob that h is the case. Let m' be the message (b, m, c) sent by Bob, informing Carol that m was sent. If the model $\mathcal{M}_0$ represents the initial situation, the messages can only be sent with m preceding m', for I assume that all messages are truthful, and the formula m is not true before m is sent. This gives:

$$\mathcal{M}_0 \xrightarrow{m} \mathcal{M}_1 \xrightarrow{m'} \mathcal{M}_2$$

What do the models look like? $\mathcal{M}_0$ is the Kripke model from Example 2.1.9. Omitting the names of the world, it looks as follows:

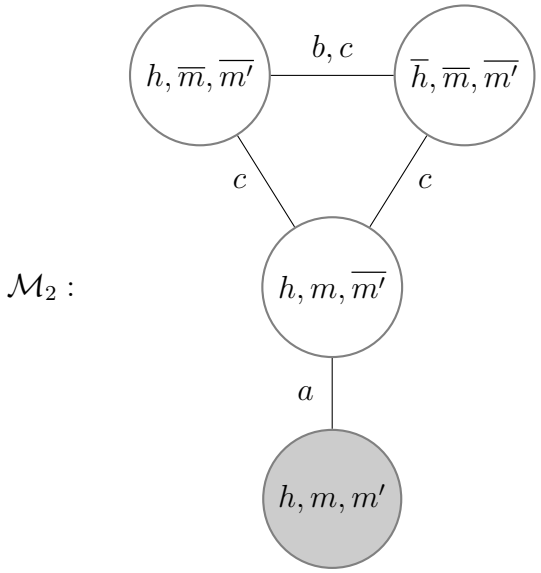


Sending message m will inform Bob about h, while Carol still considers it possible that nothing has happened. I will not only model the knowledge that Bob gains about h, but also the message itself and Bob's knowledge of it.



This model has three worlds: one where h is true and the message m is sent, one where h is true and the message m is not sent, and one where h is false and m is not sent. Since I only consider truthful communication, it is not possible that h is false and m is sent. Alice and Bob know that h is true and m is sent, therefore they do not confuse the actual world with any other world. However, Carol thinks it possible that m was not sent, and confuses the actual world with the situation where both Bob and Carol are uncertain about the value of h.

Now Bob sends Carol the message m', informing her that h is true. Alice does not know that this message is sent. I model this as follows:



In the actual world, Alice, Bob and Carol all know that h is true. Bob and Carol know that m' was sent, but Alice does not know this. Therefore she considers it possible that m' was not sent, and that Carol does not know about m and h. She confuses the actual world with the situation from model $\mathcal{M}_1$.

In order to vary the set of messages that are considered in each model, I will use vocabulary-based Kripke models. These models were introduced in [van Eijck et al., 2011]. Every vocabulary-based Kripke model has a finite vocabulary. In my set-up these vocabularies consist of the propositions and messages that are under consideration. They are the same vocabularies that I defined in the previous section. The formal definition is as follows.

3.3.2. Definition. Let a set of agents Ag, a set of propositional atoms P and a set of messages M be given. A ***vocabulary-based Kripke model*** for Ag, P, M is a tuple $\mathcal{M} = (W, R, Val, Voc, W_0)$ where W is a set of worlds, R is a function that assigns to each $a \in Ag$ an equivalence relation R_a on W, $Voc \subseteq P \cup M$ is a vocabulary of propositions and messages under consideration, Val is a function that assigns to each world in W a subset of Voc (its valuation), and $W_0 \subseteq W$ is

the set of actual worlds. I will sometimes use $\sim_a$ for R_a. Given a Kripke model $\mathcal{M}$, I will use $W^{\mathcal{M}}, R^{\mathcal{M}}, Val^{\mathcal{M}}, Voc^{\mathcal{M}}, W_0^{\mathcal{M}}$ to denote its elements.

When a message is sent, this should be modeled by a vocabulary extension combined with a knowledge update. First I will add the new message to the vocabulary of the Kripke model. It is not yet in the valuation of any world, so it is false in all worlds. Then I will use an action model to both set the truth value of the new message in the different worlds and immediately model its epistemic effects. In order to set the truth value of the new message, I need an action model that can actually change the valuation of the worlds, instead of just the relations between them. Such models are defined in [van Benthem et al., 2006]. The following definition follows the same lines.

First of all, I define a substitution that can be used to change the valuation of a world.

3.3.3. Definition. Let a set of agents Ag and a set of propositional atoms P and a set of messages M be given. A ***substitution*** over P, M is a partial function $\sigma : (P \cup M) \mapsto \{\top, \bot\}$ that assigns a new truth value to a subset of all propositions and messages. Given a valuation $Val \subseteq P \cup M$, the result of applying σ to Val is given by

$$Val \cdot \sigma := Val \setminus dom(\sigma) \cup \{x \in dom(\sigma) \mid \sigma(x) = \top\}.$$

Let $\text{SUB}_{P,M}$ be the set of all substitutions over P, M.

A substitution changes the truth value of a number of elements of a vocabulary. It leaves the truth value of the elements that are not in its domain unchanged. I will add a substitution to every event of the action model.

3.3.4. Definition. An ***action model with substitution*** for Ag, $P \cup M$ is a tuple $\mathcal{A} = (E, R, Pre, Sub, E_0)$ where E, R, Pre, E_0 are defined like the corresponding elements of an action model and $Sub : E \mapsto \text{SUB}_{P,M}$ is a function that assigns to each event a substitution over P, M.

The purpose of these action models with substitution is that the substitution of an event is applied to the valuation of all worlds matched to the event. This is reflected in the new definition of the product update:

3.3.5. Definition. Given a Kripke model $\mathcal{M}$ and an action model with substitution $\mathcal{A}$ over $Voc^{\mathcal{M}}$, the result of updating $\mathcal{M}$ with $\mathcal{A}$ is the model $\mathcal{M} \otimes \mathcal{A} = (W', R', Val', Voc', W_0')$ given by

$$\begin{aligned}
W' &:= \{(w, e) \mid w \in W^{\mathcal{M}}, e \in E^{\mathcal{A}}, \mathcal{M} \models_w Pre(e)\},\\
(w, d)R'_a(v, e) &\text{ iff } wR_a^{\mathcal{M}}v \text{ and } dR_a^{\mathcal{A}}e,\\
Val'((w, e)) &:= (Val^{\mathcal{M}} \cdot Sub^{\mathcal{A}}(e))(w),\\
Voc' &:= Voc^{\mathcal{M}},\\
W_0' &:= \{(w, e) \in W' \mid w \in W_0^{\mathcal{M}} \text{ and } e \in E_0^{\mathcal{A}}\}
\end{aligned}$$

Now I am ready to define the action model that represents the act of sending a message. It should reflect a number of properties of messages. First of all, I assume that all communication is truthful. Therefore the message may only be sent if the sender knows its contents to be true. Furthermore, all recipients of the message should get to know that the message was sent, and outsiders should not get to know this. The following action model ensures these properties.

$$\mathcal{A}_m : \boxed{e_m : [s_m]c_m, m := \top} \overset{Ag \setminus G}{\text{———}} \boxed{e_{\overline{m}} : m := \bot}$$

Here $G = s_m \cup r_m$ is the set of senders and recipients of the message. The action model has two possible events. In the first one, m is set to true so the message is sent. It has precondition $[s_m]c_m$, so the message can only be sent if the sender knows its contentss. In the second one m is set to false, so the message is not sent. he only agents who confuse the two worlds (and thus do not know whether the message is sent) are those agents that are not involved in the message.

Note that both events of the action model are in the set of actual events. This means that this action model does not determine whether the message was sent or not. It only extends the model with the possibility of sending the message, taking its content and epistemic consequences into account.

An event in an action model will only be matched with a world in a Kripke model if this world satisfies the precondition of the event. Therefore one could wonder whether the events in $\mathcal{A}_m$ will match the worlds in some Kripke model it is applied to.

Because the event $e_{\overline{m}}$ has no precondition, for every world w from the original model $\mathcal{M}$ there will be a world $(w, \overline{m})$ in the model $\mathcal{M} \otimes \mathcal{A}_m$. For the other event e_m, things are not so easy. If a world $w \in W^{\mathcal{M}}$ does not satisfy $[s_m]c_m$ then it will not match the event e_m and there will be no world (w, e_m) in the final model $\mathcal{M} \otimes \mathcal{A}_m$. This matches the intuition of the models: it is always possible not to send a message m but if the sender of m does not know its contents, then it is not possible to send it so the event representing the situation where the message is sent does not match any worlds in the Kripke model.

For the sake of brevity, I will define the result of adding a message m to a model $\mathcal{M}$ as

$$\mathcal{M} \bullet m := (W^{\mathcal{M}}, R^{\mathcal{M}}, Val^{\mathcal{M}}, Voc^{\mathcal{M}} \cup \{m\}, W_0^{\mathcal{M}}) \otimes \mathcal{A}_m.$$

I will also abbreviate (w, e_m) with (w, m) and $(w, e_{\overline{m}})$ with $(w, \overline{m})$.

The following lemma shows that this operation does not change any basic facts about the world:

3.3.6. Lemma. *For any model $\mathcal{M}$, message $m \notin Voc^{\mathcal{M}}$ and formula $\psi \in \mathcal{L}_0^{MPD}$ such that $voc(\psi) \subseteq Voc^{\mathcal{M}}$,*

$$\mathcal{M} \models_w \psi \text{ iff } \mathcal{M} \bullet m \models_{(w,\overline{m})} \psi.$$

Furthermore, if $(w, m) \in W^{\mathcal{M}\bullet m}$ then

$$\mathcal{M} \models_w \psi \text{ iff } \mathcal{M} \bullet m \models_{(w,m)} \psi.$$

Proof. A simple induction on ψ. □

The following theorem shows that in the case that m was not sent, the knowledge of the agents about basic facts does not change. Furthermore, even if m was sent the knowledge of the agents who did not receive the message does not change.

3.3.7. Theorem. *For any model $\mathcal{M}$, message $m \notin Voc^{\mathcal{M}}$ and formula $\psi \in \mathcal{L}_0^{MPD}$ such that $voc(\psi) \subseteq Voc^{\mathcal{M}}$,*

$$\mathcal{M} \models_w [a]\psi \text{ iff } \mathcal{M} \bullet m \models_{(w,\overline{m})} [a]\psi.$$

Furthermore, if $a \notin r_m$ then

$$\mathcal{M} \models_w [a]\psi \text{ iff } \mathcal{M} \bullet m \models_{(w,m)} [a]\psi.$$

Proof. Suppose $\mathcal{M} \models_w [a]\psi$. Suppose $(w, \overline{m}) \sim_a (w', x)$. Then $w \sim_a w'$ so $\mathcal{M} \models_{w'} \psi$ and by Lemma 3.3.6, $\mathcal{M} \bullet m \models_{(w',x)} \psi$. So $\mathcal{M} \bullet m \models_{(w,\overline{m})} [a]\psi$.
Suppose $\mathcal{M} \bullet m \models_{(w,\overline{m})} [a]\psi$. Suppose $w \sim_a w'$. As noted above, certainly $(w', \overline{m}) \in W^{\mathcal{M}\bullet m}$. Then because $w \sim_a w'$, it also holds that $(w, \overline{m}) \sim_a (w', \overline{m})$ so $\mathcal{M} \bullet m \models_{(w',\overline{m})} \psi$. Then by Lemma 3.3.6, $\mathcal{M} \models_{w'} \psi$. So $\mathcal{M} \models_w [a]\psi$.

Let $a \notin r_m$. Suppose $\mathcal{M} \bullet m \models_{(w,m)} [a]\psi$. Let $w \sim_a w'$. Then since $i \notin r_m$, $(w, m) \sim_a (w', \overline{m})$ so $\mathcal{M} \bullet m \models_{(w',\overline{m})} \psi$. Then by Lemma 3.3.6, $\mathcal{M} \models_{w'} \psi$. So $\mathcal{M} \models_w [a]\psi$.
Suppose $\mathcal{M} \models_w [a]\psi$. Let $(w, m) \sim_a (w', x)$. Then $w \sim_a w'$ so $\mathcal{M} \models_{w'} \psi$. Then by Lemma 3.3.6, $\mathcal{M} \bullet m \models_{(w',x)} \psi$. So $\mathcal{M} \bullet m \models_{(w,m)} [a]\psi$. □

Using this framework, I can now show formally that BCCs are unordered.

3.3.8. Theorem. *Let $\mathcal{M}, w$ be such that $\mathcal{M} \models_w m$. Let $m' = (s_m, m, a)$ and $m'' = (s_m, m, b)$. Then*

$$\mathcal{M} \bullet m' \bullet m'', ((w, m'), m'') \trianglerighteq \mathcal{M} \bullet m'' \bullet m', ((w, m''), m').$$

PROOF. Check that

$$\{(((w,x),y),((w,y),x)) \mid w \in W^{\mathcal{M}}, x \in \{m', \overline{m'}\}, y \in \{m'', \overline{m''}\}\}$$

is a bisimulation. □

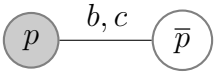
Theorem 3.3.8 and its (easy) proof illustrate how this framework can be used to formalize and prove subtle properties of message passing.

There is one other problem I have to tackle. Suppose a message is sent that mentions some other message which is not in the vocabulary of the model. Then both messages have to be added to the vocabulary: not only the message that is sent at that moment, but also the message that is mentioned in the first message. Therefore, I propose the following modeling procedure. When a message m is considered that mentions some messages of which $m_1 \preceq ... \preceq m_n$ are the ones that are not in the vocabulary of the Kripke model $\mathcal{M}$, I define the result of the ***update*** of $\mathcal{M}$ with m as

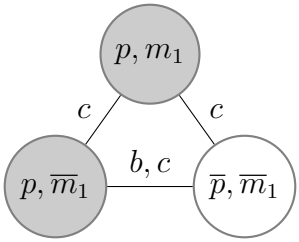
$$\mathcal{M} \odot m := \mathcal{M} \bullet m_1 \bullet m_2 \bullet ... \bullet m_n \bullet m.$$

The next example shows how this framework can be used to model the establishment of 'common knowledge of learning'. Agent b learns whether p is true from agent a, and this fact becomes common knowledge, but outsiders do not learn whether p is true from the interaction.

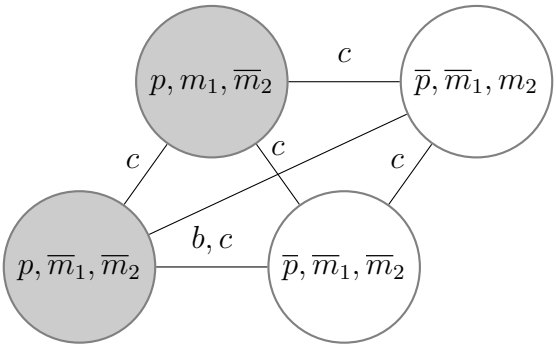
3.3.9. EXAMPLE. Consider a situation where agent a knows whether p, while agent b and c do not (and this is common knowledge). Actually, p is true. This can be represented with the following model:



Let m_1 be the message (a, p, b) and let m_2 be the message $(a, \neg p, b)$. The result $\mathcal{M} \odot m_1$ of updating with m_1:



The result $\mathcal{M} \odot m_1 \odot m_2$ of consecutively updating with m_2:



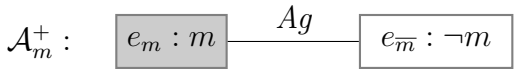
Notice that agent c confuses all worlds, since she would not receive either message m_1 or message m_2 if they were sent. On the other hand, in the worlds where m_1 or m_2 was sent, agent a and b have common knowledge of the truth value of p. Now suppose that agent a wants to create common knowledge among the three agents that a and b know the truth value of p, without revealing that truth value to agent c. Then he could send a third message m_3 of the form $(a, m_1 \vee m_2, \{a, b, c\})$ that informs the three of them that either m_1 or m_2 was sent without revealing which of the two was actually sent. When the model was updated with this third message, the resulting model would show that in those worlds where m_3 was sent, it holds that $[c]([b]p \vee [b]\neg p)$, so agent c knows that agent b knows whether p, but neither $[c]p$ nor $[c]\neg p$ hold, so agent c does not know the value of p herself.

It can be very interesting to reason about messages in a hypothetical way. For example, one could wonder whether the agents know what the epistemic consequences of sending a certain message would be. In order to express these questions I add two new constructs to the language $\mathcal{L}^{MPD}$. The formula $[\![m]\!]\varphi$ stands for "if message m is sent, φ holds". The formula $[\![\overline{m}]\!]\varphi$ stands for "if the model is extended with the possibility of sending m but it is not sent, φ holds". The semantics of these constructs is defined as follows:

$$\begin{array}{lll} \mathcal{M} \models_w [\![m]\!]\varphi & \text{iff} & \mathcal{M} \odot m \models_{(w,m)} \varphi, \\ \mathcal{M} \models_w [\![\overline{m}]\!]\varphi & \text{iff} & \mathcal{M} \odot m \models_{(w,\overline{m})} \varphi. \end{array}$$

Note that I use double brackets for modalities that express something about a different model, for example a model obtained by updating the current model with an action model, while I use single brackets for modalities that express something about different worlds in the current model, for example worlds related by an agent's relation.

As mentioned before, in the update with $\mathcal{A}_m$ I do not assume the message is actually sent. Both the world where m is sent and the world where m is not sent are actual worlds. In some situations, it is useful to denote in the model that actually the message *was* sent. For this purpose I use another action model.

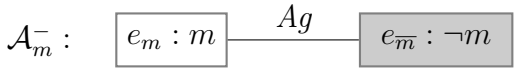


This action model divides the worlds of any Kripke model updated with it into those that satisfy m and those that do not. The worlds that satisfy m and that are actual worlds remain actual, while those that do not satisfy m become non-actual worlds. Because there are relations between e_m and $e_{\overline{m}}$ for all agents, all relations that are present in the original model are preserved. So the only thing this model does is that it makes actual worlds that do not satisfy m non-actual.

The corresponding update is defined as follows. Suppose a message m is actually sent and it mentions messages $m_1 \preceq ... \preceq m_n$ that are not in the vocabulary of $\mathcal{M}$. The result of the ***positive update*** of $\mathcal{M}$ with m is defined as follows:

$$\mathcal{M} \oplus m := \mathcal{M} \bullet m_1 \bullet ... \bullet m_n \bullet m \otimes \mathcal{A}^+_m.$$

In situations where m was actually not sent, this can also be denoted in the model. For this purpose I define yet another action model:



This model is very similar to $\mathcal{A}^+_m$, only now the worlds that do not satisfy m remain actual. Again, there is also a corresponding update. The result of the ***negative update*** of $\mathcal{M}$ with m is defined as follows:

$$\mathcal{M} \ominus m := \mathcal{M} \bullet m_1 \bullet ... \bullet m_n \bullet m \otimes \mathcal{A}^-_m.$$

With these three action models, I have set up a framework that can be used to model a large variety of message passing situations and the agents' knowledge in them. I imagine a typical modeling task as a situation where messages may be sent in a sequence of rounds. This may be the case when, for example, two agents communicate according to a set protocol. Another example is a game of poker where every player has the possibility to call, raise or fold in every round. The modeling procedure I propose is to start out with an initial model that has no messages in the vocabulary, and then gradually update the model whenever a message is sent (using $\oplus$) or could have been sent but was not (using $\ominus$).

In the next section I will show that not all possible Kripke models represent a realistic situation and I will define a class of models that do.

3.4 Models with Realistic Properties

In this section I will take a closer look at the axiomatic properties of the models I introduced. As mentioned above, I assume that all communication is truthful and reliable. This is also reflected in the update mechanism I proposed, as is shown by the following theorem.

3.4.1. THEOREM. *For any model $\mathcal{M}$ and any sequence of messages $m_1, ..., m_n \notin Voc^{\mathcal{M}}$ such that $m_1 \preceq ... \preceq m_n$, the following formulas are valid in $\mathcal{M} \bullet m_1 \bullet ... \bullet m_n$ for any m_i:*

$$m_i \to c_{m_i},$$
$$m_i \to [a]m_i \qquad \text{for all } a \in r_{m_i}$$

PROOF. I claim that for any $1 \leq i \leq n$, the above formulas hold in $\mathcal{M} \bullet m_1 \bullet ... \bullet m_i$ for all m_j with $1 \leq j \leq i$. I will prove this by induction on i. Suppose $i = 1$. I consider $\mathcal{M} \bullet m_1$. Every world in $\mathcal{M} \bullet m_1$ must be the result of matching a world from $\mathcal{M}$ with an event from $\mathcal{A}_{m_1}$. Because $e_{\overline{m_1}}$ sets the truth value of m_1 to $\bot$, the worlds matched with that event will not satisfy m_1 so they will certainly satisfy $m_1 \to c_{m_1}$ and $m_1 \to [r_{m_1}]m_1$. Now consider the other event e_{m_1}. It has precondition $[s_{m_1}]c_{m_1}$ so it satisfies c_{m_1} and thereby $m_1 \to c_{m_1}$. For the second formula I have to check that the worlds matched with e_{m_1} satisfy $[a]m_1$ for any $a \in r_m$. Take such a. Because there is no relation from e_{m_1} to $e_{\overline{m_1}}$ for agents in r_{m_1}, the only worlds that are a-related to worlds matched with e_{m_1} are other worlds matched with e_{m_1}. Because e_{m_1} sets the truth value of m_1 to $\top$, these worlds satisfy m_1. So all worlds matched with e_{m_1} satisfy $[a]m_1$ for all $a \in r_m$.

For the induction step, suppose $\mathcal{M} \bullet m_1 \bullet ... \bullet m_i$ satisfies both formulas for all m_j with $1 \leq j \leq i$. Consider $\mathcal{M} \bullet m_1 \bullet ... \bullet m_{i+1}$. With a reasoning analogous to that for the previous case I can show that the formulas hold for m_{i+1}. All that is left is to show that the formulas for $m_1, ..., m_i$ are preserved in the transition from $\mathcal{M} \bullet m_1 \bullet ... \bullet m_i$ to $\mathcal{M} \bullet m_1 \bullet ... \bullet m_{i+1}$. For the first formula this follows from Lemma 3.3.6: note that $m_j \to c_m$ is a formula from $\mathcal{L}_0^{MPD}$ that does not contain m_{i+1}, for $j \leq i$. For the second formula, note that by Lemma 3.3.6 the truth value of m_j is preserved in the update. Also, the update does not add any relations between worlds, it only possibly removes some relations. So if all a-related worlds satisfy m_j in $\mathcal{M} \bullet m_1 \bullet ... \bullet m_i$, this will also hold in $\mathcal{M} \bullet m_1 \bullet ... \bullet m_{i+1}$. Therefore both formulas are preserved for all m_j with $1 \leq j \leq i$. □

But these properties are not enough to ensure that the Kripke models are realistic. There are more subtle requirements for reasonable models, as the following example shows.

3.4.2. EXAMPLE. Consider the following model with three agents a, b and c and a message $m = (b, p, c)$:

There are two possible situations, one where m was sent and one where it was not sent, and none of the agents confuse the two situations. All communication in this model is truthful and reliable but still there is something strange about

the model: agent a knows whether the message from b to c was sent, even though she should not have received it.

It is hard to express the above property in the language $\mathcal{L}^{MPD}$: it will not do to simply state that agents that are not recipients should not know about a message, for they may have received a forward of this message and in that case they should know about it. Problems like the one in the above model would not occur if one started out with a model without messages in the vocabulary and then sequentially added new messages. Therefore, the class of models I would like to consider is the class of properly generated models:

3.4.3. Definition. A model $\mathcal{M}$ is ***properly generated*** iff there is some model $\mathcal{M}_0$ and a list of messages $m_1, ..., m_n$ such that there are no messages in the vocabulary of $\mathcal{M}_0$ and

$$\mathcal{M} \leftrightarrows \mathcal{M}_0 \bullet m_1 \bullet ... \bullet m_n.$$

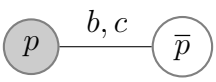
So a model is properly generated if it can be built from a model containing no messages (I call such a model an ***initial*** model) by adding messages. These are the models I consider realistic. Therefore, I want to find a procedure to check whether a Kripke model is properly generated. The rest of this section will be devoted to this task.

Consider a model $\mathcal{M}$ that is updated with a message m. As mentioned in the previous section, for every world $w \in W^{\mathcal{M}}$ there will be a world $(w, \overline{m}) \in W^{\mathcal{M} \bullet m}$. The only difference between w and $(w, \overline{m})$ is that the message m is added to the vocabulary. The relations between $\neg m$ worlds in $\mathcal{M} \bullet m$ are the same as the relations between the worlds in $\mathcal{M}$. The only difference is in the relations to and between m worlds. Therefore, if one cuts off all worlds that satisfy m and only considers the $\neg m$ worlds, this gives the original model again. This can be done with the following action model:

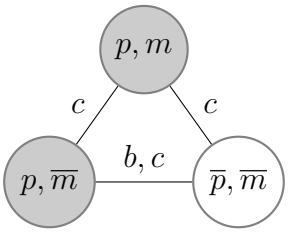
$$\mathcal{A}_{m^-} : \boxed{e : \overline{m}}$$

I will show how this works out with the following example.

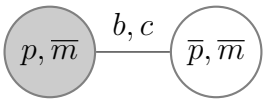
3.4.4. Example. Consider the model from Example 3.3.9 again.



Updating with $m = (a, p, b)$ gives the following result:



Now when I update with the action model $\mathcal{A}_{m^-}$ I get a model which is very much like the original, but with m in the vocabulary:



Apart from the addition of m, the third model is identical to the first one.

The following theorem shows that updating with $\mathcal{A}_{m^-}$ really gives the original model from before the update with m, if one does not consider the fact that m is now in the vocabulary.

3.4.5. Theorem. *For any model $\mathcal{M}$ such that $m \notin Voc^{\mathcal{M}}$, $\mathcal{M} \leftrightarrows_{\setminus\{m\}} \mathcal{M} \bullet m \otimes \mathcal{A}_{m^-}$.*

Proof. Let $w \in W^{\mathcal{M}}$. Then $(w, \overline{m}) \in W^{\mathcal{M}\bullet m}$ and possibly $(w, m) \in W^{\mathcal{M}\bullet m}$. But since (w, m) satisfies m if it exists, $(w, m) \notin W^{\mathcal{M}\bullet m\otimes\mathcal{A}_{m^-}}$. I define the relation Z between $W^{\mathcal{M}}$ and $W^{\mathcal{M}\bullet m\otimes\mathcal{A}_{m^-}}$ as follows.

$$\text{For any } w \in W^{\mathcal{M}}, wZ(w, \overline{m}).$$

Clearly, Z is a bisimulation if one does not consider m so $\mathcal{M} \leftrightarrows_{\setminus\{m\}} \mathcal{M} \bullet m \otimes \mathcal{A}_{m^-}$. □

With this action model, I can check whether a model is the result of an update with the message m by first "undoing" the update by updating with $\mathcal{A}_{m^-}$ and then "redoing" it by updating with m. If the result is bisimilar to the original model then I know that it is the result of the message update. I will extend this to sequences of messages. In order to do this I first need the following lemma.

3.4.6. Lemma. *For any sequence of messages $m_1, ..., m_n$ such that $m_1 \preceq ... \preceq m_n$ and for any two models $\mathcal{M}, \mathcal{N}$ such that $\mathcal{M} \leftrightarrows_{\setminus\{m_1,...,m_n\}} \mathcal{N}$,*

$$\mathcal{M} \bullet m_1 \bullet ... \bullet m_n \leftrightarrows \mathcal{N} \bullet m_1 \bullet ... \bullet m_n$$

PROOF. Let Z be a bisimulation between $\mathcal{M}$ and $\mathcal{N}$. I define a relation X between $\mathcal{M} \bullet m_1 \bullet ... \bullet m_n$ and $\mathcal{N} \bullet m_1 \bullet ... \bullet m_n$ as follows. For any two worlds $w \in W^{\mathcal{M}}$ and $v \in W^{\mathcal{N}}$ and any sequence $x = x_1, ..., x_n$ where $x_i = m_i$ or $x_i = \overline{m_i}$,

$$(...(w, x_1), x_2), ..., x_n) X (...(v, x_1), x_2), ..., x_n) \text{ iff } wZv$$

Note that the question of whether (w, x) exists depends on whether $\mathcal{M} \models_w [s_{m_1}]c_{m_1}$ if $x_1 = m_1$, and whether $\mathcal{M} \models_{(w,x_1)} [s_{m_2}]c_{m_2}$ if $x_2 = m_2$, etcetera. Similarly for (v, x) and $\mathcal{N}$. But because $m_1 \preceq ... \preceq m_n$, these things only depend on the propositions and messages that are true in w and in v (which are the same because wZv) and the earlier messages. So (w, x) exists iff (v, x) exists. So X is total. It is clear from the definition of message update that X is a bisimulation. □

Now I can characterize the class of properly generated models using the action model $\mathcal{A}_{m^-}$ and the message update:

3.4.7. THEOREM. *A model $\mathcal{M}$ is properly generated iff there is an order $m_1, ..., m_n$ listing all messages in the vocabulary of $\mathcal{M}$ such that $m_1 \preceq ... \preceq m_n$ and*

$$\mathcal{M} \leftrightarrows \mathcal{M} \otimes \mathcal{A}_{m_n^-} \otimes ... \otimes \mathcal{A}_{m_1^-} \bullet m_1 \bullet ... \bullet m_n.$$

PROOF. $\Rightarrow$: Suppose $\mathcal{M}$ is properly generated. Then there is some initial model $\mathcal{M}_0$ and a list of messages $m_1, ..., m_n$ such that $\mathcal{M} \leftrightarrows \mathcal{M}_0 \bullet m_1 \bullet ... \bullet m_n$.
By repeated use of Theorem 3.4.5, I have

$$\mathcal{M}_0 \leftrightarrows_{\setminus\{m_n,...,m_1\}} \mathcal{M} \otimes \mathcal{A}_{m_n^-} \otimes ... \otimes \mathcal{A}_{m_1^-}.$$

Then by Lemma 3.4.6,

$$\mathcal{M} \leftrightarrows \mathcal{M} \otimes \mathcal{A}_{m_n^-} \otimes ... \otimes \mathcal{A}_{m_1^-} \bullet m_1 \bullet ... \bullet m_n.$$

$\Leftarrow$: Suppose there is such an order $m_1, ..., m_n$. Let $\mathcal{N}$ be the model like $\mathcal{M} \otimes \mathcal{A}_{m_n^-} \otimes ... \otimes \mathcal{A}_{m_1^-}$, but with $m_1, ..., m_n$ not in the vocabulary. Clearly,

$$\mathcal{M} \otimes \mathcal{A}_{m_n^-} \otimes ... \otimes \mathcal{A}_{m_1^-} \leftrightarrows_{\setminus\{m_1,...,m_n\}} \mathcal{N}$$

so by Lemma 3.4.6,

$$\mathcal{M} \otimes \mathcal{A}_{m_n^-} \otimes ... \otimes \mathcal{A}_{m_1^-} \bullet m_1 \bullet ... \bullet m_n \leftrightarrows \mathcal{N} \bullet m_1 \bullet ... \bullet m_n.$$

But because $m_1, ..., m_n$ are all the messages in the vocabulary of $\mathcal{M}$, I conclude that $\mathcal{N}$ is an initial model. This implies that $\mathcal{M} \otimes \mathcal{A}_{m_n^-} \otimes ... \otimes \mathcal{A}_{m_1^-} \bullet m_1 \bullet ... \bullet m_n$ is properly generated, and then so is $\mathcal{M}$. □

3.5 Axiomatization

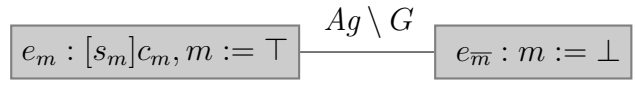
I have added two modalities $[\![m]\!]$ and $[\![\overline{m}]\!]$ to the language $\mathcal{L}^{MPD}$. In [van Benthem et al., 2006] a technique is developed for translating a language with action modalities to epistemic PDL. I will use the same technique to show that these three modalities do not increase the expressive power of $\mathcal{L}^{MPD}$. For each formula containing a modality I will give a reduction axiom that shows that the formula with the modality is equivalent to a formula without it. For the Boolean cases, these reduction axioms look as follows:

$$
\begin{array}{lcll}
[\![m]\!]p & \leftrightarrow & [s_m]c_m \to p & \\
[\![m]\!]m' & \leftrightarrow & [s_m]c_m \to m' & m' \neq m \\
[\![m]\!]m & \leftrightarrow & \top & \\
[\![m]\!]\neg\phi & \leftrightarrow & \neg[\![m]\!]\phi & \\
[\![m]\!](\phi_1 \vee \phi_2) & \leftrightarrow & [\![m]\!]\phi_1 \vee [\![m]\!]\phi_2 &
\end{array}
$$

$$
\begin{array}{lcll}
[\![\overline{m}]\!]p & \leftrightarrow & p & \\
[\![\overline{m}]\!]m' & \leftrightarrow & m' & m' \neq m \\
[\![\overline{m}]\!]m & \leftrightarrow & \bot & \\
[\![\overline{m}]\!]\neg\phi & \leftrightarrow & \neg[\![\overline{m}]\!]\phi & \\
[\![\overline{m}]\!](\phi_1 \vee \phi_2) & \leftrightarrow & [\![\overline{m}]\!]\phi_1 \vee [\![\overline{m}]\!]\phi_2 &
\end{array}
$$

The reduction axioms for formulas containing epistemic programs (the PDL modalities α, corresponding to relations in the model) are more complicated. This is because when a relation is followed in the Kripke model with an epistemic program, the same relation can only be followed in the model which is the result of the message update if this relation is not removed by the update.

Recall that the message updates correspond to an update with the following action model:



A relation will be present in the result of the update if it is both in the original model and in the action model. So I have to check whether the epistemic program can be executed in the updated model by checking whether it can be executed both in the original model and in the action model "concurrently".

For all epistemic programs, I will compute an epistemic program that is the equivalent of the original program together with a concurrent step in the action model $\mathcal{A}_m$. With $T_{xy}(\alpha)$ I mean the program that is the equivalent of doing α in the original model and concurrently moving from state e_x to state e_y in the action model. I define it inductively as follows:

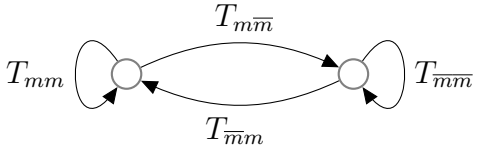
$$
\begin{array}{lcl}
T_{mm}(a) & := & ?[s_m]c_m; a; ?[s_m]c_m \\
T_{m\overline{m}}(a) & := & \begin{cases} ?\bot & \text{if } a \in r_m \\ ?[s_m]c_m; a & \text{otherwise} \end{cases} \\
T_{\overline{m}m}(a) & := & \begin{cases} ?\bot & \text{if } a \in r_m \\ a; ?[s_m]c_m & \text{otherwise} \end{cases} \\
T_{\overline{m}\overline{m}}(a) & := & a
\end{array}
$$

$$
\begin{array}{lcl}
T_{mm}(?\psi) & := & ?([s_m]c_m \wedge \psi) \\
T_{m\overline{m}}(?\psi) & := & ?\bot \\
T_{\overline{m}m}(?\psi) & := & ?\bot \\
T_{\overline{m}\overline{m}}(?\psi) & := & ?\psi
\end{array}
$$

$$
\begin{array}{l}
T_{mm}(\alpha_1; \alpha_2) := (T_{mm}(\alpha_1); T_{mm}(\alpha_2)) \cup (T_{m\overline{m}}(\alpha_1); T_{\overline{m}m}(\alpha_2)) \\
T_{m\overline{m}}(\alpha_1; \alpha_2) := (T_{mm}(\alpha_1); T_{m\overline{m}}(\alpha_2)) \cup (T_{m\overline{m}}(\alpha_1); T_{\overline{m}\overline{m}}(\alpha_2)) \\
T_{\overline{m}\overline{m}}(\alpha_1; \alpha_2) := (T_{\overline{m}\overline{m}}(\alpha_1); T_{\overline{m}\overline{m}}(\alpha_2)) \cup (T_{\overline{m}m}(\alpha_1); T_{m\overline{m}}(\alpha_2)) \\
T_{\overline{m}m}(\alpha_1; \alpha_2) := (T_{\overline{m}\overline{m}}(\alpha_1); T_{\overline{m}m}(\alpha_2)) \cup (T_{\overline{m}m}(\alpha_1); T_{mm}(\alpha_2))
\end{array}
$$

$$
\begin{array}{lcl}
T_{mm}(\alpha_1 \cup \alpha_2) & := & T_{mm}(\alpha_1) \cup T_{mm}(\alpha_2) \\
T_{m\overline{m}}(\alpha_1 \cup \alpha_2) & := & T_{m\overline{m}}(\alpha_1) \cup T_{m\overline{m}}(\alpha_2) \\
T_{\overline{m}m}(\alpha_1 \cup \alpha_2) & := & T_{\overline{m}m}(\alpha_1) \cup T_{\overline{m}m}(\alpha_2) \\
T_{\overline{m}\overline{m}}(\alpha_1 \cup \alpha_2) & := & T_{\overline{m}\overline{m}}(\alpha_1) \cup T_{\overline{m}\overline{m}}(\alpha_2)
\end{array}
$$

The final case is the reduction for α^*. Note that the action model can be seen as the following automaton:



Then the epistemic program giving all finite paths through the action model starting in e_m and ending in e_m is:

$$T_{mm}{}^*(T_{m\overline{m}}T_{\overline{m}\overline{m}}{}^*T_{\overline{m}m}T_{mm}{}^*)^*.$$

Similarly, if I take e_m as start state and $e_{\overline{m}}$ as final state, I get:

$$T_{mm}{}^*T_{m\overline{m}}T_{\overline{m}\overline{m}}{}^*(T_{\overline{m}m}T_{mm}{}^*T_{m\overline{m}}T_{\overline{m}\overline{m}}{}^*)^*.$$

For $e_{\overline{m}}$ as start and as stop state:

$$T_{\overline{m}\overline{m}}{}^*(T_{\overline{m}m}T_{mm}{}^*T_{m\overline{m}}T_{\overline{m}\overline{m}}{}^*)^*.$$

And finally, if I take $e_{\overline{m}}$ as start state and e_m as stop state:

$$T_{\overline{m}\overline{m}}{}^*T_{\overline{m}m}T_{mm}{}^*(T_{m\overline{m}}T_{\overline{m}\overline{m}}{}^*T_{\overline{m}m}T_{mm}{}^*)^*.$$

All in all I get the following recipe for transforming an epistemic expression of the form α^*:

$$\begin{array}{lll}
T_{mm}(\alpha^*) := & (T_{mm}(\alpha))^*; (T_{m\overline{m}}(\alpha); (T_{\overline{m}\overline{m}}(\alpha))^*; T_{\overline{m}m}(\alpha); (T_{mm}(\alpha))^*)^*, \\
T_{m\overline{m}}(\alpha^*) := & (T_{mm}(\alpha))^*; T_{m\overline{m}}(\alpha); (T_{\overline{m}\overline{m}}(\alpha))^*; (T_{\overline{m}m}(\alpha); (T_{mm}(\alpha))^*; \\
& T_{m\overline{m}}(\alpha); (T_{\overline{m}\overline{m}}(\alpha))^*)^*, \\
T_{\overline{m}\overline{m}}(\alpha^*) := & (T_{\overline{m}\overline{m}}(\alpha))^*; (T_{\overline{m}m}(\alpha); (T_{mm}(\alpha))^*; T_{m\overline{m}}(\alpha); (T_{\overline{m}\overline{m}}\alpha))^*)^*, \\
T_{\overline{m}m}(\alpha^*) := & (T_{\overline{m}\overline{m}}(\alpha))^*; T_{\overline{m}m}(\alpha); (T_{mm}(\alpha))^*; (T_{m\overline{m}}(\alpha); (T_{\overline{m}\overline{m}}(\alpha))^*; \\
& T_{\overline{m}m}(\alpha); (T_{mm}(\alpha))^*)^*
\end{array}$$

Now I can give the reduction axioms for the case of epistemic programs:

$$\begin{array}{lcl}
[\![m]\!][\alpha]\phi & \leftrightarrow & [T_{mm}(\alpha)][\![m]\!]\phi \wedge [T_{m\overline{m}}(\alpha)][\![\overline{m}]\!]\phi \\
[\![\overline{m}]\!][\alpha]\phi & \leftrightarrow & [T_{\overline{m}\overline{m}}(\alpha)][\![\overline{m}]\!]\phi \wedge [T_{\overline{m}m}(\alpha)][\![m]\!]\phi
\end{array}$$

This gives:

3.5.1. Theorem. *The language $\mathcal{L}^{MPD}$ and the language $\mathcal{L}^{MPD}$ with message modalities added have the same expressive power.*

Proof sketch. Take any formula φ from $\mathcal{L}^{MPD}$ with message modalities. Any message modality in φ can be replaced with an equivalent subformula that contains no message modalities. The correct equivalent subformulas are prescribed by the reduction axioms given above. This way, I can find for any formula that contains message modalities an equivalent formula that does not contain them. Therefore, the message modalities do not add expressive power to $\mathcal{L}^{MPD}$. □

3.6 Related Work

The work presented in this chapter was inspired by the wish to incorporate explicit messages in Dynamic Epistemic Logic (DEL). I will clarify what the added value of my approach is compared to the usual DEL as in [Baltag and Moss, 2004, van Benthem et al., 2006, van Ditmarsch et al., 2006]. In the usual DEL, there is no mention of any messages and the only atoms in the models are propositions. The models can be updated with so-called action models, of which my message update is a special case. In my approach I have tailored an action model for a specific kind of group messages with a sender and a set of recipients. This is very useful in modeling since it is no longer up to the user of the framework to come up with the right action model: this is automatically "generated" when defining the message. This way, I make a step towards formalizing the modeling procedure which makes it easier and less error-prone.

I have combined DEL with the vocabulary expansion proposed in [van Eijck et al., 2011] and used this to introduce messages explicitly in the models. This has the great advantage that it is possible to model agents who reason about messages that have been sent and even messages about other messages. This allows for constructions like forward, acknowledgement, BCC recipients etcetera.

In my approach every model has a vocabulary of propositions and messages that the agents are aware of. The vocabulary of a Kripke model can be viewed as a global awareness function, indicating the set of propositions and messages that the agents are aware of across the model. A more extended study of awareness in a similar setting can be found in [Fagin and Halpern, 1988, van Ditmarsch and French, 2011]. There, a more subtle notion of awareness is presented, where different agents may be aware of different vocabularies in different worlds.

My work can be compared to interpreted systems as presented in e.g. [Fagin et al., 1995]. There, the focus is on a global state that is constructed by combining local states of the agents. In this set up, two global states are related for an agent if the corresponding local states of that agent are equivalent. In my approach, there is no clear distinction between one agent's and another agent's information. One possible such distinction would be to say that an agent's local state is her "inbox" of messages she sent or received up to that moment. Then one would somehow also have to incorporate the messages forwarded to the agent.

The idea of time is clearly incorporated in interpreted systems. In my framework this is less explicit: I can show how the model evolves over time by doing a sequence of message updates, but once these updates have been done the only information that is preserved in the model is whether the message has been sent at some point in time, not when it was sent exactly or an ordering between them. Of course there is the vocabulary embedding relation $\prec$, but this only partially orders the messages. This has the advantage of keeping the model simple, and in a lot of applications the exact ordering between messages is not so relevant.

3.7 Conclusion

I have shown how epistemic models can be used to represent the influence of message passing on the knowledge of agents. The models presented in this chapter directly show the agent's knowledge using relations between possible worlds. The models are finite and I have given an axiomatization. A nice property of this approach is that the models can be generated automatically given a sequence of messages that have been sent.

This system has the curious property that agents are affected by an update with messages that are not addressed to them: they consider the fact that such a message was sent possible. The history-based system of Parikh and Ramanujam has the same property, as does the process of updating with S5 action models for group announcements (see, e.g., [Baltag and Moss, 2004]).

In some situations this property is perfectly realistic, for example in a game where in every new round the agents know which new messages may be sent. However, when modeling everyday communication it is less realistic: when two people are communicating and a third person does not know what they are communicating about the third person usually thinks any message is possible, and does not have a specific possible message in mind.

One possible solution that comes to mind is to give every agent a personal set of messages she is aware of. However, this does not solve the problem. For consider an agent a that does not know whether p is the case, and suppose a message m is sent to some other agent b, informing her that p is the case. Then, even if a is not aware of m, something changes in the model that a can notice: after m was sent a must hold it for possible that the other agent b has learnt something about p.

Look at this informally. How can an agent i ever know for sure that another agent j does *not* know whether p? Suppose initially $[i](\neg[j]p \wedge \neg[j]\neg p)$. Suppose i holds it for possible that some other agent k knows whether p. In other words, $\langle i\rangle([k]p \vee [k]\neg p)$ holds. How can this situation persist? How can i be sure that k does not send a secret message (k, p, j) or $(k, \neg p, j)$?

One possible solution would be to always start from initial models where $[i](\neg[j]p \wedge \neg[j]\neg p)$ does not hold, for any i, j, p. However, this has the disadvantage of blowing up the size of the initial models. In Chapters 5 and 6 I will present two different approaches that immediately take all possible messages into account, instead of using a limited vocabulary of messages. This is more realistic in some situations, but it will become clear that this comes with a price in the form of infinitely large models. Especially in game-theoretic situations where there is a limited number of messages or signals that can be sent in each round, or when the agents are following some known protocol consisting of a limited number of possible messages, the approach given in this chapter is a lot more appropriate and efficient.

Chapter 4

Logic of Information Flow on Communication Channels

4.1 Introduction

In this chapter, I present a framework for modeling communication and knowledge that is very general and can be adapted to the natural needs of various situations. The approaches presented in Chapters 3, 5 and 6 are tailored towards specific situations. This is very convenient when modeling exactly such a situation, but if those approaches are not applicable then the approach presented in this chapter will be fit for modeling almost any other situation involving communication and knowledge. Furthermore, in this chapter I also give an explicit treatment of protocols which broadens the perspective to include a great number of issues that come up in practice.

As a running example, consider the following situation. The 1999 'National Science Quiz' of *The Netherlands Organisation for Scientific Research (NWO)* [1] had the following question:

> *Six friends each have one piece of gossip. They start making phone calls. In every call they exchange all pieces of gossip that they know at that point. How many calls at least are needed to ensure that everyone knows all six pieces of gossip?*

To reason about the information flow in such a scenario, I want to take into account the following issues: the messages that the agents possess (e.g. secrets), the knowledge of the agents, the dynamics of the system in terms of information passing (e.g. telephone calls), the underlying communication channels (e.g. the network of landlines) and the protocol the agents follow (e.g. a method to exchange all pieces of gossip). I will combine all these different aspects in an

[1] For a list of references about the problem, cf. [Hurkens, 2000].

approach that is a new combination of Dynamic Epistemic Logic (DEL) and Interpreted Systems (IS).

Interpreted Systems, introduced by [Parikh and Ramanujam, 1985] and [Fagin et al., 1995] independently, are mathematical structures that combine history-based temporal components of a system with epistemic ones (defined in terms of *local states* of the agents). This framework is convenient when modeling knowledge development based on the given temporal development of a system. In IS, the epistemic structure of a system is generated from the temporal structure in a uniform way. However, the generation of temporal structures is not specified in the framework.

A different perspective on the dynamics of multi-agent systems is provided by DEL [Gerbrandy and Groeneveld, 1997, Baltag and Moss, 2004]. The main focus of DEL is not on the temporal structure of the system but on the epistemic impact of events as the agents perceive them. The development of a system through time is essentially generated by executing action models as discussed in Chapter 3 and 7. The epistemic relations in the initial static model and in the action models are not generated uniformly as in IS. Instead, they are designed by hand. How to obtain a reasonable initial model that fits the scenario to be modeled is not always clear. For real life applications it can be hard to find the correct initial model. Finding the correct action models that correspond to epistemic events can be even harder, as is observed in [Dechesne and Wang, 2007].

Much has been said about the comparison of the two frameworks, based on the observation that certain temporal developments of the system in IS can be generated by sequences of DEL updates on static models (see, e.g., [van Benthem et al., 2009a, Hoshi and Yap, 2009, Hoshi, 2009]). In this chapter, I will demonstrate further benefits of combining the two approaches by presenting a framework where epistemic relations are generated by matching local states and a history of observations as in IS, while keeping the flexibility of explicit actions as in DEL approaches.

The puzzle of the telephone calls was briefly discussed in [van Ditmarsch, 2000, Ch. 6.6] within the original DEL framework. Van Benthem [van Benthem, 2002] raised the research question whether the communication network can be made explicit in DEL. An early proposal to fill in this line of research can be found in [Roelofsen, 2005]. Communication channels in an IS framework made their appearance in [Parikh and Ramanujam, 2003]. In [Pacuit and Parikh, 2007, Apt et al., 2009] the information passing on so-called communication graphs or interaction structures is adressed, where messages are modeled as either atomic propositions or Boolean combinations of atomic propositions. In [Wang et al., 2009] a PDL-style DEL language is developed that allows explicit specification of protocols.

This chapter is organized as follows. I introduce the logic $\mathcal{L}_\iota^{Ag,N}$ in Section 4.2. Section 4.3 relates the logic to the standard DEL and IS approaches. Section 4.4 introduces a modeling method and illustrates this method by a study of variations

on the puzzle that was mentioned above. The final section concludes and lists future work.

4.2 An Adaptable Logic for Communication, Knowledge and Protocols

In this section I will present a flexible logic that can be adapted to the situation at hand. I will first give the language with its intuitive meaning. Then I will define the states on which this language is to be interpreted, together with its formal semantics.

4.2.1 Language

Let Ag be a finite set of agents, N a finite set of atomic notes and Act a finite set of basic actions. Later on, I will give each action an internal structure that defines its meaning, but for now the actions may be considered to be atomic objects.

I define *net* to be a hypergraph of agents in Ag, representing the communication network. It is a set of subsets of Ag, just like in the approach presented in [Apt et al., 2009]. Each subset represents a possible set of recipients of a single message. For example, if $net = \{\{a, b\}, \{a, b, c\}\}$ then the communication network allows for private communication between agent a and b and for group communication between agents a, b and c. This rules out private communication between b and c or a and c.

The set $P_{Ag,M,Act}$ of basic propositions is defined as

$$p := has_a n \mid com(G) \mid past(\bar{\alpha}) \mid future(\bar{\alpha}),$$

where $a \in Ag$, $n \in N$, $G \subseteq Ag$ and $\bar{\alpha} = \alpha_1; ...; \alpha_n$ with $\alpha_1, ..., \alpha_n \in Act$.

The intended meaning of these propositions is as follows. The proposition $has_a n$ means that agent a possesses note n. This is a piece of information that he may send to other agents. The proposition $com(G)$ means that G is a communication channel, so a group message to the group G is in accordance with the communication network. The proposition $past(\bar{\alpha})$ means that the sequence of actions that happened most recently is $\bar{\alpha}$. Finally, the proposition $future(\bar{\alpha})$ means that the sequence of actions $\bar{\alpha}$ could be executed now, in accordance with the current protocol.

Using these propositons, I define the formulas of $\mathcal{L}_\iota^{Ag,N}$ as follows:

$$\begin{aligned} \varphi &::= \top \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle\pi\rangle\phi \mid C_G\varphi, \\ \pi &::= \alpha \mid \varepsilon \mid \delta \mid \pi_1; \pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^*, \end{aligned}$$

where $p \in P_{Ag,M,Act}$, $G \subseteq Ag$, $\alpha \in Act$ and ε, δ are constants for the empty sequence and deadlock, respectively. I define Π as the set of all possible protocols π.

The intended meaning of the formulas is as follows. The meaning of $\top$ and the constructs $\neg$ and $\wedge$ is as usual. $C_G\phi$ expresses "the agents in group G have common knowledge of ϕ". A difference between this language and the one presented in Chapter 3 is that now, $\langle\pi\rangle\phi$ expresses "the protocol π can be executed, and at least one execution of π yields a state where φ holds". So instead of expressing that φ holds in a world considered possible by an agent, this formula now expresses that φ holds in a state that is a possible result of the protocol π. The protocol π is built from actions as the relations in Chapter 3 are built from the agent's epistemic relations.

As mentioned above, I will give each action an ***internal structure***. This internal structure is given for each $\alpha \in Act$ as a tuple of the following form:

$$\iota(\alpha) := \langle G, \phi, N_1, ...N_{|Ag|}, \rho\rangle$$

Here $G \subseteq Ag$ is the group of agents that can observe α. ϕ is a formula of $\mathcal{L}_\iota^{Ag,N}$ that *does not* contain any modalities of the form $\langle\pi\rangle$. Moreover, it is the precondition that should hold in order for α to be executable. I define $Obs(\iota(\alpha)) = G$ and $Pre(\iota(\alpha)) = \phi$. Additionally, $Pos(\iota(\alpha)) = \langle N_1, ...N_{|Ag|}, \rho\rangle$ is the postcondition that should hold after α has been executed. For every agent a, N_a is the set of notes that get delivered to a by action α. Finally, $\rho \in \Pi \cup \{\#\}$ gives the protocol that the agents are going to follow after execution of α. If $\rho = \#$, then the agents should keep following the current protocol. If $\rho = \pi$ for some $\pi \in \Pi$ then they should change their protocol to π. I will assume that an agent can observe any action by which he receives some note. The converse does not hold: agents may also observe actions by which no notes are delivered to them. This happens for example when an agent knows that some other agent receives a message containing a certain note, but he does not get to know the contents of the note himself.

Note that by excluding the preconditions of the form $\langle\pi\rangle\varphi$ I limit the interdependence of actions. This prevents problems when for example an action would be mentioned in its own precondition. Even with this constraint I can still express a lot of useful preconditions. For example, for action α, $future(\alpha)$ is allowed as a precondition meaning that α can be executed only when it is allowed by the current protocol.

As usual, I define $\bot$, $\phi \vee \psi$, $\phi \to \psi$ and $[\pi]\phi$ as the abbreviations of $\neg\top$, $\neg(\neg\phi \wedge \neg\psi)$, $\neg\phi \vee \psi$ and $\neg\langle\pi\rangle\neg\phi$ respectively. Moreover, I use the following additional abbreviations:

$$\begin{aligned}
K_a\phi &:= C_{\{a\}}\phi \\
has_a N &:= \textstyle\bigwedge_{n\in N} has_a n \\
dhas_G N &:= \textstyle\bigwedge_{n\in N} \bigvee_{a\in G} has_a n \\
com(net) &:= \textstyle\bigwedge_{G\in net} com(G) \wedge \bigwedge_{G\notin net} \neg com(G) \\
\pi^n &:= \underbrace{\pi;\pi;...;\pi}_{n \text{ times}} \\
\Sigma\Pi' &:= \textstyle\bigcup_{\pi\in\Pi'} \pi \text{ where } \Pi' \subset \Pi \text{ is finite.}
\end{aligned}$$

Here $K_a\phi$ means that agent a knows ϕ, $dhas_G N$ expresses that the messages from N are in distributed possession of the agents in G and $com(net)$ specifies the communication channels in the network.

By having both the *has* and the K operator in the language, I can make the distinction between knowing about a message and knowing about its content. $K_a has_b n \wedge \neg has_a n$ and $K_a has_b n \wedge has_a n$ can express the *de dicto* and *de re* reading of knowing that b has a message, respectively. For example, let n be the hiding place of Bin Laden, then $K_{CIA} has_{Al-Qaeda} n \wedge \neg has_{CIA} n$ expresses that CIA knows that Al-Qaeda knows the hiding place, which is, however, a secret to CIA.

4.2.2 Semantics

In order to interpret the basic propositions in $P_{Ag,M,Act}$ I let the finer structure of the basic propositions correspond with a finer structure in the states, replacing the traditional valuation in Kripke structures used in DEL-approaches.

4.2.1. Definition. A state for $\mathcal{L}_\iota^{Ag,N}$ is defined as a tuple:

$$s := \langle net, N_1^I, ..., N_{|Ag|}^I, \bar{\alpha}, N_1, ..., N_{|Ag|}, \pi \rangle.$$

Here *net* is the communication graph, $\bar{\alpha}$ is the history of actions that have been executed, for every $a \in Ag$ N_a gives the set of notes he possesses and π gives the protocol the agents are following. I also include for every agent $a \in Ag$ the set N_a^I which is the set of notes the agents had in the ***initial state***, which was the state of the systems before the actions in $\bar{\alpha}$ were executed. Given a state s, I use $N(s)(a)$ to denote N_a, the information set of agent a. I use $N^I(s)(a)$ to denote N_a^I, the initial information set of agent a. I use $Net(s) := net$ for the communication graph, $H(s) := \alpha$ for the action history and $Prot(s) := \pi$ for the protocol.

Intuitively, each state represents a past temporal development of the system with its constraint for the future actions. Note that the past is linear ($\bar{\alpha}$ is a single sequence of actions), while the future can be branching (the protocol π may allow several possible sequences of actions). From the initial information sets I can construct the initial state of the system before any actions were executed. For a state s as in the previous definition, this is defined as

$$Init(s) := \langle net, N_1^I, ..., N_{|Ag|}^I, \epsilon, N_1^I, ..., N_{|Ag|}^I, (\Sigma Act)^* \rangle.$$

The initial state has an empty action history, and the information sets of the agents are identical to the initial information sets. Also, no protocol has been set so the protocol is $(\Sigma Act)^*$, which allows all sequences of actions. Note that for any state s, the result of executing the history of past actions on $Init(s)$ should be s.

I will interpret the formulas of $\mathcal{L}_\iota^{Ag,N}$ on the states defined above. However, in order to give the semantics for *future*$(\bar{\alpha})$ I need a way to check whether a sequence of actions complies with a certain protocol. Also, in order to give the semantics for $\langle\pi\rangle$ I need to be able to compute the remainder of the protocol after the action has been executed, so I know what the new protocol is. For this purpose I will use the ***input derivative*** and the ***output function*** (cf. [Brzozowski, 1964, Conway, 1971]).

I start out with the output function. This function returns ϵ if the protocol π can be executed by doing no action, and δ otherwise. It is defined as follows:

$$\begin{array}{rclcrcl} o(\varepsilon) & := & \varepsilon, & & o(\delta) & := & \delta, \\ o(\alpha) & := & \delta, & & o(\pi \cup \pi') & := & o(\pi) \cup o(\pi'), \\ o(\pi;\pi') & := & o(\pi); o(\pi'), & & o(\pi^*) & := & \epsilon. \end{array}$$

Given a protocol π and an action α, the remainder of π after executing α is the input derivative $\pi\backslash\alpha$ given by:

$$\begin{array}{rclcrcll} \varepsilon\backslash\alpha & := & \delta, & \delta\backslash\alpha & := & \delta, & \\ \alpha\backslash\alpha & := & \epsilon, & \beta\backslash\alpha & := & \delta & (\alpha \neq \beta), \\ (\pi \cup \pi') \cup \alpha & := & \pi\backslash\alpha \cup \pi'\backslash\alpha, & & & & \\ (\pi;\pi')\backslash\alpha & := & ((\pi\backslash\alpha);\pi') \cup (o(\pi);(\pi'\backslash\alpha)), & & & & \\ (\pi^*)\backslash\alpha & := & \pi\backslash\alpha;\pi^*. & & & & \end{array}$$

Let $\pi\backslash(\alpha_0;\alpha_1;\ldots;\alpha_n) = (\pi\backslash\alpha_0)\backslash\alpha_1\ldots\backslash\alpha_n$. Using these definitions and the axioms of Kleene algebra I can syntactically derive the remaining protocol after executing a sequence of basic actions. For example:

$$\begin{array}{rl} (\alpha \cup (\beta;\gamma))^*\backslash\beta & = (\alpha\backslash\beta \cup (\beta;\gamma)\backslash\beta);(\alpha \cup (\beta;\gamma))^* \\ & = (\delta \cup (\varepsilon;\gamma));(\alpha \cup (\beta;\gamma))^* \\ & = \gamma;(\alpha \cup (\beta;\gamma))^*. \end{array}$$

Note that in general it does not hold that $\bar{\beta};(\pi\backslash\bar{\beta}) = \pi$.

Let $A(\pi)$ be the set of sequences of actions that comply with the protocol π. It is defined as follows:

$$\begin{array}{l} A(\delta) = \emptyset \qquad A(\varepsilon) = \{\epsilon\} \qquad A(\alpha) = \{\alpha\} \\ A(\pi;\pi') = \{\bar{\alpha};\bar{\beta} \mid \bar{\alpha} \in A(\pi), \bar{\beta} \in A(\pi')\} \\ A(\pi \cup \pi') = A(\pi) \cup A(\pi') \\ A(\pi^*) = \{\bar{\alpha}_1;\ldots;\bar{\alpha}_n \mid \bar{\alpha}_1,\ldots,\bar{\alpha}_n \in A(\pi)\} \end{array}$$

In [Conway, 1971], the following is shown:

4.2.2. Lemma. $A(\pi\backslash\bar{\alpha}) = \{\bar{\beta} \mid \bar{\alpha};\bar{\beta} \in A(\pi)\}$.

This shows that the input derivative truly computes the remainder of the protocol after executing some basic action.

Just like [Cohen and Dam, 2007, Apt et al., 2009], I will give the truth value of $\mathcal{L}_\iota^{Ag,N}$ formula on single states instead of pointed Kripke models as is usual in DEL. The interpretation of epistemic formulas depends on a relation $\sim_a^x$ between states, which I will define later.

Given a state $s = \langle net, N_1^I, ..., N_{|Ag|}^I, \bar{\alpha}, N_1, ..., N_{|Ag|}, \pi\rangle$, the semantics of $\mathcal{L}_\iota^{Ag,N}$ is defined as follows.

$$\begin{array}{lll}
s \models has_a(n) & \text{iff} & n \in N_a \\
s \models com(G) & \text{iff} & G \in net \\
s \models past(\bar{\beta}) & \text{iff} & \bar{\beta} \text{ is a suffix of } \bar{\alpha} \\
s \models future(\bar{\beta}) & \text{iff} & \pi \backslash \bar{\beta} \neq \delta \\
s \models \neg\varphi & \text{iff} & s \not\models \varphi \\
s \models \varphi_1 \wedge \varphi_2 & \text{iff} & s \models \varphi_1 \text{ and } s \models \varphi_2 \\
s \models \langle\pi\rangle\varphi & \text{iff} & \exists s' : s[\![\pi]\!]s' \text{ and } s' \models \varphi \\
s \models C_G\varphi & \text{iff} & \forall s' : s \sim_G^x s' \text{ implies } s' \models \varphi
\end{array}$$

Here $\sim_G^x$ is the reflexive transitive closure of $\bigcup_{a\in G} \sim_a^x$. As noted above, the relation $\sim_a^x$ is the knowledge relation for agent a and it will be more formally defined later.

The protocols π function as state changers. Each protocol describes a transition to a new state in the following way:

$$\begin{array}{rll}
s[\![\varepsilon]\!]s' & \text{iff} & s = s' \\
s[\![\delta]\!]s' & & \text{never} \\
s[\![\beta]\!]s' & \text{iff} & s \models Pre(\iota(\beta)) \text{ and } s' = s|_{Pos(\iota(\beta))} \\
s[\![\pi_1;\pi_2]\!]s' & \text{iff} & \exists s'' : s[\![\pi_1]\!]s'' \text{ and } s''[\![\pi_2]\!]s' \\
s[\![\pi_1 \cup \pi_2]\!]s' & \text{iff} & s[\![\pi_1]\!]s' \text{ or } s[\![\pi_2]\!]s' \\
s[\![(\pi_1)^*]\!]s' & \text{iff} & \exists n : s[\![\underbrace{\pi_1;\pi_1;...;\pi_1}_{n \text{ times}}]\!]s'
\end{array}$$

Given $Pos(\iota(\beta)) = \langle N'_1, \ldots, N'_{|Ag|}, \rho\rangle$, $s|_{Pos(\iota(\beta))}$ is the result of executing action β at s. It is defined as

$$s|_{Pos(\iota(\beta))} = \langle net, N_1^I, \ldots, N_{|Ag|}^I, \bar{\alpha};\beta, N_1 \cup N'_1, \ldots, N_{|Ag|} \cup N'_{|Ag|}, f(\rho)\rangle,$$

$$\text{where } f(\rho) = \begin{cases} \pi\backslash\beta \text{ if } \rho = \# \\ \pi' \quad \text{if } \rho = \pi' \end{cases}.$$

So I add the action β to the sequence of past actions, I add for each agent a the notes he received by β and I change the protocol to a new protocol π' if this is prescribed by β, or to the remainder of the old protocol after executing β if no new protocol is dictated.

Now I will define the epistemic relation of an agent a between states. This relation depends on the observational power of the agents, which may vary in different situations. Therefore I represent it as a relation $\sim_a^{obs}$, where *obs* stands

for the observational power of the agents. A state s is said to be *consistent* if $Init(s)[\![H(s)]\!]s$. It is easy to see that for any s, $Init(s)$ is always consistent. Note that I can actually omit the current information sets $N(s)$ in the definition of a state, and compute them by applying the actions in $H(s)$ to $N^I(s)$, thus only generating consistent states. I keep the current information sets in the definition of the state in order to simplify the notation and to evaluate basic propositions more efficiently.

I define that $s \sim_a^{obs} s'$ if and only if the following conditions are met:

consistency s and s' are consistent.

local initialization $N^I(s)(a) = N^I(s')(a)$,

local history $H(s)|_a^{obs} = H(s')|_a^{obs}$, where *obs* is the *type of observational power* of agents.

The type of observational power of the agents defines how the agents observe the history. In other words, it defines their local history $H(s)|_a^{obs}$. Many definitions of $H(s)|_a^{obs}$ are possible, giving the agents different observational powers. This is one of the things that make this framework so flexible and allow for adaptation to different situations. Several reasonable definitions are:

1. $H(s)|_a^{set} = \{\alpha \text{ appearing in } H(s) \mid a \in Obs(\iota(\alpha))\}$ as in [Apt et al., 2009] and in Chapter 5 and 6. In this set-up, the agents are aware of the actions they can observe but not of the ordering between these actions.

2. $H(s)|_a^{1st}$ is the subsequence of $H(s)$ consisting of the first occurrence of each $\alpha \in H(s)|_a^{set}$ as in [Baskar et al., 2007]. In this set-up, the agents are aware of the ordering of the first occurrence of the actions they can observe.

3. $H(s)|_a^{asyn}$ is the subsequence of $H(s)$ consisting of all the occurrences of each $\alpha \in H(s)|_a^{set}$, as in *asynchronous* systems (cf., e.g., [Shilov and Garanina, 2002]). In this set-up, the agents are aware of all occurrences of the actions they can observe and the ordering between them.

4. $H(s)|_a^{\tau}$ is the sequence obtained from $H(s)$ by replacing each occurrence of $\alpha \notin H(s)|_a^{set}$ by τ, as in *synchronous* systems with prefect recall (cf., e.g., [van der Meyden and Shilov, 1999]). In this set-up, the agents are aware of all occurrences of the actions they can observe and they are also aware of the number of actions that have been happened that they cannot observe, and of the order between the actions they can observe and the actions they cannot observe. They do not get to know which actions that they cannot observe have happened.

It is clear from the above definition that $\sim_a^{obs}$ is an equivalence relation and the following holds:

4.2.3. LEMMA. $\sim_a^{\tau} \subseteq \sim_a^{asyn} \subseteq \sim_a^{1st} \subseteq \sim_a^{set}$.

So the $\sim^{\tau}$ relation is the smallest relation, thereby giving the agents the greatest amount of knowledge, and the $\sim_a^{set}$ relation is the largest, giving the agents only little knowledge.

I call the semantics defined by $\sim_a^{obs}$ the *obs-semantics*, and denote the corresponding satisfaction relation as $\models^{obs}$.

Recall that the agents can always observe the actions that change their information set. This implies the following lemma.

4.2.4. LEMMA. *For any consistent state s, $s \sim_a^{obs} s'$ implies $N(s)(a) = N(s')(a)$, where $obs \in \{set, asyn, 1st, \tau\}$.*

PROOF. By Lemma 4.2.3, $s \sim_a^{obs} s'$ implies $s \sim_a^{set} s'$ for all $obs \in \{set, asyn, 1st, \tau\}$. Therefore I only need to prove the claim for $obs = set$. Suppose $s \sim_a^{set} s'$. Then by the definition of $\sim_a^{set}$, $N(Init(s))(a) = N(Init(s'))(a)$ and $H(s)|_a^{set} = H(s')|_a^{set}$. So at s and s' agent a initially had the same messages and has observed the same actions. Since agents can always observe the actions that change their information set, this implies that the same message passing actions relevant to a have happened in s and s'. Since the actions can only add notes to the information sets of the agents and never delete notes from them, it does not matter how often or in which order those actions have been executed. Therefore the information sets of agent a in s and s' are identical. □

By using different semantics in different situations, I can vary the observational power of the agents as is required. By constructing actions that match the situation at hand, I can also vary the exact properties of the communicative events. I will now define some useful basic actions with their internal structure. These actions correspond to communicative events that often come up in practice.

In order to simplify the presentation, I will omit the explicit mentioning of the internal structure map ι. So I will use $Obs(\alpha)$ for $Obs(\iota(\alpha))$ etcetera. Recall that the internal structure of an action α is a tuple

$$\iota(\alpha) := \langle G, \phi, N_1, ...N_{|Ag|}, \rho \rangle$$

such that $N_a = \emptyset$ for $a \notin Obs(\alpha)$. The following table lists some basic actions. In Section 4.4 I will use these as building blocks for more complex actions.

α :	$Obs(\alpha)$:	$Pre(\alpha)$:	$Pos(\alpha)$:
$send^a_G(N)$	$G \cup \{a\}$	$com(G \cup \{a\}) \wedge future(\alpha) \wedge has_a N$	$N_b := N_b \cup N, \rho = \#$ $(b \in G)$
$share_G(N)$	G	$com(G) \wedge future(\alpha) \wedge dhas_G N$	$N_b := N_b \cup N, \rho = \#$ $(b \in G)$
$sendall^a_G$	$G \cup \{a\}$	$com(G \cup \{a\}) \wedge future(\alpha)$	$N_b := N_b \cup N_a, \rho = \#$ $(b \in G)$
$shareall_G$	G	$com(G) \wedge future(\alpha)$	$N_b := \bigcup_{a \in G} N_a, \rho = \#$ $(b \in G)$
$inform^a_G(\phi)$	$G \cup \{a\}$	$K_a \phi$	$\rho = \#$
$exinfo(\phi)$	Ag	ϕ	$\rho = \#$
$exprot(\pi)$	Ag	$\top$	$\rho = \pi$

In the rightmost column of table I have left out from the postconditions the sets of notes of the agents that do not change, in order to save space.

The first group of actions are communicative actions that are done by the agents. These actions must abide by the communication channels and the protocol, which is enforced by having $com(Obs(\alpha)) \wedge future(\alpha)$ in the precondition. $send^a_G(N)$ is the action that a sends the set of notes N to the group G. Apart from respecting the channels and the protocol, the precondition $has_a N$ enforces that agent a should possess the notes he wants to send. The postcondition of $send^a_G(N)$ expresses that the messages in N get added to the message sets of the agents in G. $share_G(N)$ shares the messages from N within the group G. A precondition is that the messages from N are already distributed knowledge in the group. $sendall^a_G$ differs from $send^a_G(N)$ in the fact that a sends *all* the notes that he has. Similarly for $shareall_G$. $inform^a_G(\phi)$ is the group announcement by a of an arbitrary formula ϕ within $G \cup \{a\}$. The precondition for this action is that agent a knows that ϕ holds. Since all agents know that the execution of this action would only be possible if ϕ would hold, all agents who can observe the action know that ϕ holds at the moment it is announced. This way knowledge of ϕ is created among the members of G.

The second group of actions are public announcements that do not respect the channels or the protocol. They model the information that is given to the agents by some external authority. $exinfo(\phi)$ models the public announcement of a formula ϕ. The only precondition of this announcement is that ϕ should hold. The postcondition is empty. Again, knowledge of ϕ is created by the fact that the agents know that the action can only be done if ϕ holds. $exprot(\pi)$ announces the protocol π that the agents are supposed to follow in the future. ts postcondition changes the protocol to π and knowledge of the protocol is created by the fact that all agents observe the announcement.

4.3 Comparison with IS and DEL

The results in this section relate my logic to IS and DEL approaches. Theorem 4.3.1 shows that by the semantics of $\mathcal{L}_\iota^{Ag,N}$, an interpreted system is implicitly generated from a single state. Together with Theorem 4.3.1, Theorem 4.3.3 demonstrates that compared to DEL, my approach models actions in a very powerful and concise manner.

I will compare my approach to IS first. In the following I only consider consistent states.

Given a state s with action history $H(s) = \alpha_1\alpha_2...\alpha_n$, I define the history of s as the sequence $his(s) = s_0s_1...s_n$ where $s_0 = Init(s)$, $s_n = s$ and for all $1 \leq k \leq n$, $s_{k-1}[\![\alpha_k]\!]s_k$. Clearly then $s_0s_1 \dots s_k = his(s_k)$ for any $k \leq n$.

Given some type of semantics *obs*, let $ExpT^{obs}$ be the Interpreted System given by $\{H, \rightarrow_\alpha, \{R_i \mid i \in Ag\}, V\}$, where

- $H = \{his(s) \mid s \text{ is consistent}\}$,
- $\langle s_0 \dots s_n\rangle \rightarrow_\alpha \langle s_0 \dots s_n s_{n+1}\rangle$ iff $s_n[\![\alpha]\!]s_{n+1}$,
- $\langle s_0 \dots s_n\rangle R_i \langle s'_0 \dots s'_m\rangle$ iff $s_n \sim_i^{obs} s'_m$,
- $V(\langle s_0 \dots s_n\rangle)(p) = \top$ iff $s_n \models^{obs} p$, where $p \in P_{Ag,M,Act}$.

This is a straightforward adaptation of my logic to the IS framework. The language $\mathcal{L}_\iota^{Ag,N}$ can be seen as a fragment of Propositional Dynamic Logic (PDL) with basic actions taken from $Act \cup Ag$. Then the C_G operator corresponds to $(\Sigma G)^*$. Let $\models_{PDL}$ denote the ususal semantics of this fragment. The following theorem follows easily:

4.3.1. THEOREM. *For any formula $\varphi \in \mathcal{L}_\iota^{Ag,N}$ and for each consistent $\mathcal{L}_\iota^{Ag,N}$-state s:*

$$s \models^{obs} \varphi \text{ iff } ExpT^{obs}, hist(s) \models_{PDL} \varphi.$$

This result shows that when I abstract away the inner structure of basic propositions and actions, then the logic can be seen as a PDL language interpreted on ISs that are generated in a particular way in accordance with some constraints.

Next, I will compare my work to standard DEL. Consider the following DEL language $\mathcal{L}_{DEL}$:

$$\phi := \top \mid p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid [\![\mathcal{A}, e]\!]\phi \mid C_G\phi$$

Here p is taken from a set of basic propositions P, $G \subseteq Ag$ and $\mathcal{A}$ is an action model, as defined in Chapter 2, with e as its designated action. The formula $[\![\mathcal{A}, e]\!]\phi$ holds in $\mathcal{M}, w$ for some Kripke model $\mathcal{M}$ and $w \in W^{\mathcal{M}}$ iff ϕ holds in $\mathcal{M} \otimes \mathcal{A}, (w, e)$.

I would like to see if a translation is possible from $\mathcal{L}_\iota^{Ag,N}$ to DEL. Such a translation would go from the actions of $\mathcal{L}_\iota^{Ag,N}$ to the action models of DEL. A protocol π would then correspond to a sequence of action models. The first barrier

in the way of such a translation is the fact that the $*$ operator allows for arbitrarily long sequences of actions, while there is no such operator on modalities of action models in DEL. Therefore, I will consider the star-free fragment of $\mathcal{L}_\iota^{Ag,N}$.

However, it turns out that even without the $*$ operator it is not possible to find a translation for all kinds of semantics ($set, 1st$, etcetera). To see why this is true, recall the following result from [van Benthem et al., 2009a].

4.3.2. Theorem ([van Benthem et al., 2009a]). *If we see $[\![\mathcal{A}, e]\!]$ as a basic action modality in the semantics of the PDL language, then for any formula $\varphi \in \mathcal{L}_{DEL}$ and for any model $\mathcal{M}$ and state $w \in W^{\mathcal{M}}$:*

$$\mathcal{M}, w \models_{DEL} \phi \text{ iff } Forest(\mathcal{M}, \mathbb{A}), (w) \Vdash_{PDL} \phi$$

Here $\mathbb{A}$ is the set of action models and $Forest(\mathcal{M}, \mathbb{A})$ is the IS generated by executing all possible sequences of action models in $\mathbb{A}$ on $\mathcal{M}$.

Using this theorem, I will now show that the effects of actions in $\mathcal{L}_\iota^{Ag,N}$ cannot, in general, be simulated by action models.

4.3.3. Theorem. *There is* no *DEL-model $\mathcal{M}$ such that for all consistent $\mathcal{L}_\iota^{Ag,N}$-states s there is some $w \in W_{\mathcal{M}}$ that satisfies for all formulas $\varphi \in \mathcal{L}_\iota^{Ag,N}$:*

$$s \models \varphi \text{ iff } \mathcal{M}, w \models_{DEL} \phi.$$

Proof. Suppose there was such $\mathcal{M}$. Then by Theorem 4.3.1 and 4.3.2,

$$(ExpT^{obs}, hist(s)) \leftrightarrows (Forest(\mathcal{M}, \mathbb{A}), (w)),$$

where $\leftrightarrows$ is the bisimulation for transitions labeled with $Act \cup Ag$. In [van Benthem et al., 2009a] it is shown that any model of the form $Forest(\mathcal{M}, \mathbb{A})$ must satisfy the property of ***perfect recall***. This property states that if the agents cannot distinguish two sequences of actions $\bar{\alpha}; \alpha$ and $\bar{\beta}; \beta$ then they cannot distinguish $\bar{\alpha}$ and $\bar{\beta}$. But $ExpT^{obs}$ does not satisfy this property for $obs \in \{set, 1st, asyn\}$. For example, if γ is some action that b cannot observe then $send_b^a(N); \gamma \sim_b^{obs} \gamma; send_b^a(N)$, but $send_b^a(N) \not\sim_b^{obs} \gamma$. So the set-, $1st$- and $asyn$-semantics cannot be translated to a DEL model. □

4.4 Applications

4.4.1 Common Knowledge

This framework gives an interesting perspective on common knowledge. It may not be surprising that common knowledge cannot be reached without public communication [Halpern and Moses, 1990]. I first focus on asynchronous semantics.

One might think that achieving common knowledge becomes easier if the agents can publicly agree on a common protocol before the communication is limited to non-public communication. However, in the case of asynchronous semantics common knowledge still cannot be achieved, even if the agents can publicly agree on a protocol. Recall that I say an action α respects the communication channels if $Pre(\alpha) \models com(Obs(\alpha))$.

4.4.1. THEOREM. *For any state s with $Ag \notin Net(s)$, any protocol π containing only actions that respect the communication channels, any $\varphi \in \mathcal{L}_\iota^{Ag,N}$ and any sequence of actions $\bar{\alpha}$:*

$$s \models^{asyn} \langle exprot(\pi)\rangle(\neg C_{Ag}\varphi \to \neg\langle\bar{\alpha}\rangle C_{Ag}\varphi)$$

PROOF. Let $s[\![exprot(\pi)]\!]t$ and suppose $t \models^{asyn} \neg C_{Ag}\varphi$. Towards a contradiction, let $\bar{\alpha}$ be the minimal sequence of actions such that $t \models^{asyn} \langle\bar{\alpha}\rangle C_{Ag}\varphi$. Let $\bar{\alpha} = \bar{\beta};\alpha$, $t[\![\bar{\beta}]\!]u$ and $u[\![\alpha]\!]v$. Since $Ag \notin Net(s)$ and α respects the communication channel, $Obs(\alpha) \neq Ag$ so there exists $a \notin Obs(\alpha)$. Then $H(u)|_a^{asyn} = H(v)|_a^{asyn}$ so $u \sim_a^{asyn} v$. Since $\bar{\alpha}$ was minimal, $u \not\models^{asyn} C_{Ag}\varphi$. But then $u \models^{asyn} \neg K_a C_{Ag}\varphi$ so $v \models^{asyn} \neg K_a C_{Ag}\varphi$. So $v \not\models^{asyn} C_{Ag}\varphi$. This contradicts my assumption, so there cannot be such $\bar{\alpha}$. So $s \models^{asyn} \langle exprot(\pi)\rangle(\neg C_{Ag}\varphi \to \neg\langle\bar{\alpha}\rangle C_{Ag}\varphi)$. □

Essentially, even if the agents agree on a protocol beforehand, the agents that cannot observe the final action of the protocol will never know whether this final action has been executed and thus common knowledge is never established. This is because in the asynchronous semantics, there is no sense of time. If there would be some kind of clock and the agents would agree to do an action on every "tick", the agents would be able to establish common knowledge. This is exactly what I try to achieve with the τ-semantics. Here every agent observes a "tick" the moment some action is executed. This way, they can agree on a protocol *and* know when it is finished. I will show examples of how this can result in common knowledge in the discussion of the telephone call scenario.

Here I will first investigate what happens in τ-semantics if the agents *cannot* publicly agree on a protocol beforehand. I will show that in this case they cannot reach common knowledge of basic formulas. I start out with a lemma stating that actions preserve the agent's relations.

4.4.2. LEMMA. *For any two states s and t and any action α, if $s \sim_i^\tau t$ and there are s', t' such that $s[\![\alpha]\!]s'$ and $t[\![\alpha]\!]t'$ then $s' \sim_i^\tau t'$.*

PROOF. Suppose $s \sim_i^\tau t$. Then $H(s)|_i^\tau = H(t)|_i^\tau$. Suppose $i \in Obs(\alpha)$. Then $H(s')|_i^\tau = (H(s)|_i^\tau;\alpha) = (H(t)|_i^\tau;\alpha) = H(t')|_i^\tau$. Suppose $i \notin Obs(\alpha)$. Then $H(s')|_i^\tau = (H(s)|_i^\tau;\tau) = (H(t)|_i^\tau;\tau) = H(t')|_i^\tau$. So $s' \sim_i^\tau t'$. □

This result may seem counter-intuitive, since for example a public announcement action may give the agents new information and thus destroy their epistemic relations. However, in my framework I model the new knowledge introduced by communicative actions by the fact that these actions would not be possible in states that do not satisfy the precondition of the action. In this lemma I assume that there are s', t' such that $s[\![\alpha]\!]s'$ and $t[\![\alpha]\!]t'$. This means that s and t both satisfy the preconditions of α, so essentially no knowledge that distinguishes s and t is introduced by α.

Let $\mathcal{L}_{bool}$ be the following fragment of $\mathcal{L}_\iota^{Ag,N}$:

$$\phi ::= has_i m \mid com(G) \mid \neg\phi \mid \phi_1 \wedge \phi_2$$

It is trivial to show that any action that does not change the agents' message sets or the protocol does not change the truth value of these basic formulas:

4.4.3. Lemma. *Let α be an action that does not change the agents' message sets or the protocol. For any $\phi \in \mathcal{L}_{bool}$ and any state s: $s \models \phi \leftrightarrow \langle\alpha\rangle\phi$.*

Combining the properties of the actions from the previous lemma, I call an action $dummy(G)$ to be a ***dummy action*** for a group of agents G if it has the precondition $com(G) \wedge future(dummy(G))$, it does not change the message sets of the agents or the protocol and $Obs(dummy(G)) = G$. An example of dummy action is $inform^i_G(\top)$. One could see it as "idle talk".

4.4.4. Theorem. *Let A be a set of basic actions respecting the communication channels such that for any agent a there is a dummy action $dummy(G)$ such that $a \notin G \subseteq Ag$. Let s be a state such that $Ag \notin Net(s)$ and it is common knowledge at s that the protocol is $\pi = (\Sigma A)^*$ (any action in A is allowed). Then for any $\phi \in \mathcal{L}_{bool}$ and any sequence of actions $\bar{\alpha}$,*

$$s \models^\tau \neg C_{Ag}\phi \rightarrow \neg\langle\bar{\alpha}\rangle C_{Ag}\phi$$

Proof. Suppose towards a contradiction that $s \models \neg C_I\phi$ and there is a minimal sequence $\bar{\alpha}$ such that $s \models^\tau \langle\bar{\alpha}\rangle C_{Ag}\phi$. Let $\bar{\alpha} = \bar{\beta};\alpha$ and let $a \notin Obs(\alpha)$. Such a always exists since $Ag \notin Net(s)$. Let $dummy(G)$ be a dummy action such that $a \notin G$. Let $s[\![\bar{\beta}]\!]u$. Since $\bar{\alpha}$ is minimal, $u \models^\tau \neg C_{Ag}\phi$, so there is a $\sim_{Ag}$-path from u to a world t such that $t \not\models^\tau \phi$. Since it is common knowledge that any action in A is possible, $dummy(G)$ can be executed at any world on the path from u to t. By lemma 4.4.2 $dummy(G)$ preserves the relations between states so there are states u', t' such that $u[\![dummy(G)]\!]u'$, $t[\![dummy(G)]\!]t'$ and $u' \sim_{Ag} t'$. Also, since $t \not\models^\tau \phi$ and by lemma 4.4.3, $t' \not\models^\tau \phi$. So u' *not* $\models^\tau C_{Ag}\phi$. This means that if $dummy(G)$ would be executed in state u, then $C_{Ag}\phi$ would not hold in the resulting state.

Let $u[\![dummy(G)]\!]u'$ and $u[\![\alpha]\!]v$. Because $a \notin G$, a cannot see the difference between executing $dummy(G)$ and α: $H(u')|^\tau_a = (H(u)|^\tau_a;\tau) = H(v)|^\tau_a$ so $u' \sim^\tau_a v$.

But I just showed that $u' \not\models^{\tau} C_{Ag}\phi$, so then $v \not\models^{\tau} C_{Ag}\phi$. But this contradicts my assumption that $\bar{\beta};\alpha$ induced common knowledge of ϕ. □

Before turning to the specific scenario of the telephone calls, I propose the following general modeling method:

1. Select a set of suitable actions *Act* with internal structures to model the communicative events in the scenario.

2. Design a single state as the *real world* to model the initial setting, i.e., $\langle net, N_1, ..., N_{|Ag|}, \bar{\alpha}, N_1, ..., N_{|Ag|}, (\Sigma A)^*\rangle$ where *net* models the communication network and N_a models the information possessed by agent a.

3. Translate the informal assumptions of the scenario into formulas φ and protocols π in $\mathcal{L}_{\iota}^{Ag,N}$.

4. Use $exinfo(\varphi)$ and $exprot(\pi)$ to make the assumptions and the protocol common knowledge.

I will demonstrate how I can use this method to model the telephone call scenario. Let me first recall the scenario: in a group of people, each person has one secret. They can make private telephone calls amongst themselves in order to communicate these secrets. The original puzzle concerns the minimal number of telephone calls needed to ensure everyone gets to know all secrets.

I start out by selecting a set of suitable actions that fit the scenario. I define them as follows.

$$\begin{aligned} call_b^a &:= shareall_{\{a,b\}} \\ message_b^a &:= sendall_{\{b\}}^a \end{aligned}$$

Here $call_b^a$ is the call between agents a and b in which they share all the notes (or secrets) they possess. Later on I will also be interested in what happens if the agents can only leave voicemail messages instead of making two-way calls. For this purpose I use $message_b^a$, where agent a sends all secrets he possesses to agent b. Let $A = \bigcup_{a,b\in Ag} call_b^a \cup \bigcup_{a,b\in Ag} message_b^a$.

Next, I define the information sets of the agents. For every agent a, I define his set of notes as $N_a = \{s_a\}$, where s_a is his secret. Let S be the set of all secrets. The communication network allows for pairwise communication between the agents. I define it as $Net = \{\{a, b\} \mid a, b \in Ag\}$. Then the initial state is

$$s_I := \langle Net, \{s_1\}, ..., \{s_{|Ag|}\}, \varepsilon, \{s_1\}, ..., \{s_{|Ag|}\}, (\Sigma A)^*\rangle.$$

I want to vary the communicative powers of the agents in different situations. Therefore I will define different protocols that restrict the actions the agents can execute. I define $\pi_{call} := (\bigcup_{a,b\in Ag} call_b^a)^*$, $\pi_{mail} := (\bigcup_{a,b\in Ag} message_b^a)^*$ as the protocols where the agents can only make telephone calls or only send voicemails, respectively.

In order to reason about the number of calls the agents need to make to reach their goal, I will use the following abbreviations:

$$\begin{array}{rcl} \Diamond^{\leq n}\phi & := & \langle \bigcup_{k\leq n}(\Sigma A)^k\rangle\phi \\ \Diamond^{min(n)}\phi & := & \Diamond^{\leq n}\phi \wedge \neg\Diamond^{\leq n-1}\phi \end{array}$$

$\Diamond^{\leq n}\phi$ expresses that a state where ϕ holds can be reached by sequentially executing at most n actions from A. $\Diamond^{min(n)}\phi$ expresses that n is the minimal such number. Note that A does not contain any actions that change the protocol, therefore the formulas express whether the agents can achieve ϕ with the current protocol. Note that the temporal operator $\Diamond$ (sometimes called F) of IS approaches (e.g. [Pacuit and Parikh, 2007]) can be defined by $\langle(\Sigma A)^*\rangle$ while $\Diamond^{\leq n}$ serves as a generalization of the *arbitrary announcement* that is added to DEL in [Ågotnes et al., 2009].

Then the following result states that exactly $2|Ag| - 4$ calls are necessary to make sure every agent knows all secrets:

4.4.5. Lemma. *For any* $obs \in \{set, 1st, asyn, \tau\}$,

$$s_I \models^{obs} \langle exprot(\pi_{call})\rangle\Diamond^{min(2|Ag|-4)} \bigwedge_{a\in Ag} has_a S.$$

A proof of this proposition is given in [Hurkens, 2000]. The protocol given there is the following: pick a group of four agents 1 ... 4 and let 4 be their informant. Let agent 4 call all other agents, then let the four agents communicate all their secrets within their group and let all other agents call agent 4 again. In my framework this can be expressed as follows:

$$call_5^4; ...; call_{|Ag|}^4; call_2^1; call_4^3; call_3^1; call_4^2; call_5^4; ...; call_{|Ag|}^4$$

Now I turn to the question that arises when the agents cannot make direct telephone calls, but they can only leave voicemail messages. This means that any agent can tell the secrets he knows to another agent, but he cannot in the same call also learn the secrets the other agent knows. How many voicemail messages would the agents need in this case?

The agents could use $message_b^a; message_a^b$ to mimic each $call_b^a$, which gives

$$s_I \models^{obs} \langle exprot(\pi_{mail})\rangle\Diamond^{\leq 4|Ag|-8} \bigwedge_{a\in Ag} has_a S.$$

However, they can do much better, as the following lemma shows.

4.4.6. Lemma. *For any* $obs \in \{set, 1st, asyn, \tau\}$,

$$s_I \models^{obs} \langle exprot(\pi_{mail})\rangle\Diamond^{min(2|Ag|-2)} \bigwedge_{a\in Ag} has_a S.$$

PROOF. Consider the following protocol:

$$message_2^1; message_3^2; ...; message_{|Ag|}^{|Ag|-1}; message_1^{|Ag|}; message_2^{|Ag|}; ...; message_{|Ag|-1}^{|Ag|}.$$

Clearly, this results in all agents knowing all secrets. The length of this protocol is $2|Ag| - 2$. I claim that this protocol is minimal. To see why this claim holds, first observe that there has to be one agent who is the first to learn all secrets. For this agent to exist all other agents will first have to make at least one call to reveal their secret to someone else. This is already $|Ag| - 1$ calls. The moment that agent learns all secrets, since he is the first, all other agents do not know all secrets. So each of them has to receive at least one more call in order to learn all secrets. This also takes $|Ag| - 1$ calls which brings the total number of calls to $2|Ag| - 2$. □

As the above results show, it is possible to make sure all agents know all secrets. However, in these results the secrets are not common knowledge yet, since the agents do not know that everyone knows all secrets. I will investigate whether common knowledge of all secrets can be established. I will assume that prior to the start of the protocol, the distribution of the secrets is common knowlege. For this purpose I use the following abbreviation:

$$SecDis_{Ag} := \bigwedge_{a \in Ag} (has_a s_a \wedge \bigwedge_{b \neq a} \neg has_b s_a)$$

If there are only three agents, then achieving common knowledge of all secrets is possible by making telephone calls:

4.4.7. LEMMA. *If $|Ag| \leq 3$ then for some $n \in \mathbb{N}$:*

$$s_I \models^\tau \langle exinfo(SecDis_{Ag}); exprot(\pi_{call})\rangle \Diamond^{\leq n} C_{Ag} \bigwedge_{a \in Ag} has_a S.$$

PROOF. For $|Ag| < 3$ the proof is trivial. Suppose $|Ag| = 3$, say $Ag = \{1, 2, 3\}$. A protocol that results in the desired property is $call_2^1; call_3^2; call_1^2$. After execution of this protocol all agents know all secrets, and agent 2 knows this. Also, since agent 1 learned the secret of agent 3 from agent 2, he knows that agent 2 and 3 must have communicated after the last time he spoke to agent 2, so agent 3 must know the secret of agent 1. Regarding agent 3, he knows agent 2 has all secrets the moment he communicated with agent 2, and he observed a τ when agent 2 called agent 1 after that. Since there are only three agents, agent 3 can deduce that agent 1 and 2 communicated so he knows agent 1 knows all secrets. Since all agents can reason about each other's knowledge, it is common knowledge that all agents have all secrets. □

I do not extend this result to the case with more than three agents. If there are more than three agents, agents that are not participating in the phone call will never know which of the other agents are calling, which makes it much harder to establish common knowledge.

Now imagine a situation where the agents are beforehand allowed to publicly announce a specific protocol they are going to follow which is more complex than just the set of actions they can choose from. Then, in the τ-semantics, it is possible to reach common knowledge:

4.4.8. PROPOSITION. *There is a protocol π of call actions such that*

$$s_I \models^{\tau} \langle exinfo(SecDis_{Ag})\rangle\langle exprot(\pi)\rangle\Diamond^{\leq n} C_{Ag} \bigwedge_{a\in Ag} has_a S$$

PROOF. Let π be the protocol given in the proof of proposition 4.4.5. Since each agent observes a τ at every communicative action, they can all count the number of communicative actions that have been executed and they all know when the protocol has been executed. So at that moment, it will be common knowledge that everyone has all secrets. □

This shows the use of the ability to communicate about the future protocol and not only about the past and present. There are many more situations where announcing the protocol is very important, for example in the puzzle of 100 prisoners and a light bulb [Dehaye et al., 2003] and in many situations in distributed computing.

4.5 Conclusion

In this chapter I proposed an expressive framework that combines properties from dynamic epistemic logic and interpreted systems. The framework is very flexible and it can be adapted to almost any situation that concerns communication and knowledge. I specifically include the communication network in my set-up, which allows for reasoning about the network and about the agents' knowledge of the network. I showed how this framework can be used to model communication by applying it to the example with the telephone calls mentioned in the introduction of this chapter.

The framework is very flexible in modeling different observational powers of agents and various communicative actions. For example, the communicative action that is used in [Pacuit and Parikh, 2007], "a gets b's information without b noticing this", can be modeled as $\alpha = download^a_b$ with $Obs(\alpha) = \{a\}$, $Pre(\alpha) = com(\{a, b\})$ and a postcondition containing $N_a := N_a \cup N_b$. Because of the freedom in the design of the actions and observational powers, this framework can facilitate the comparison of different approaches with different assumptions.

Chapter 5

Common Knowledge in Email Communication

5.1 Introduction

In the previous chapters I have presented a number of models for the knowledge of agents in some message passing scenario. These models relied on a number of assumptions that made them more applicable to certain situations, but they could usually be applied to a wide range of problems. In this chapter, I will focus on one specific instance of message passing, namely email communication.

Email is by now a prevalent form of communication. From the point of view of distributed programming it may look as just another instance of multicasting - one agent sends a message to a group of agents. However, such features as forwarding and the *blind carbon copy* (BCC) make it a more complex form of communication.

The reason is that each email implicitly carries epistemic information concerning (among others) common knowledge within the group involved in it of the fact that it was sent. As a result forwarding leads to nested common knowledge and typically involves different groups of agents at each level. In turn, the BCC feature results in different information gain by the regular recipients and the BCC recipients. In fact, in Section 5.7 I show that the BCC feature is new from an epistemic point of view.

To be more specific, suppose that an agent a forwards a message m to a group G. Then the group $G \cup \{a\}$ consisting of the sender and the recipients of m acquires (among other knowledge) common knowledge of the fact that m was sent. Next, suppose that an agent a sends a message m to a group G with a BCC to a group B. Then the group $G \cup \{a\}$ acquires common knowledge of m, while each member of B separately acquires with the sender of m common knowledge of the fact that the group $G \cup \{a\}$ acquires common knowledge of m.

Combining forward and BCC, satisfaction of the epistemic formulas $C_{A_1}...C_{A_k}m$ of arbitrary depth can be realized, where C_A stands for 'the group A has common

knowledge of'. Furthermore, this combination can lead to a, usually undesired, situation in which a BCC recipient of an email reveals his status to others by using the *reply-all* feature. In general, a chain of forwards of arbitrary length can reveal to a group of agents that an agent was a BCC recipient of the original email. This shows that the email exchanges, as studied here, are essentially different from multicasting.

Epistemic consequences of email exchanges are occasionally raised by researchers in various contexts. For instance, the author of [Babai, 1990] mentions 'some issues of email ethics' by discussing a case of an email discussion in which some researchers were not included (and hence could not build upon the reported results).

Another example is the following recent quotation from a blog in which the writers call for a boycott of a journal XYZ: "We are doing our best to make the misconduct of the Editors-in-Chief a matter of common knowledge within the [...] community in the hope that everyone will consider whatever actions may be appropriate for them to adopt in any future associations with XYZ".

When studying email exchanges a natural question arises: what are their knowledge-theoretic consequences? To put it more informally: after an email exchange took place, who knows what? Motivated by the above blog entry I could also ask: can sending emails to more and more new recipients ever create common knowledge?

To be more specific, consider the following example, to which I shall return later.

5.1.1. Example. Assume the following email exchange involving four people, Alice, Bob, Clare and Daniel:

- Alice and Daniel got an email from Clare,
- Alice forwarded it to Bob,
- Bob forwarded Alice's email to Clare and Daniel with a BCC to Alice,
- Alice forwarded the last email to Clare and Daniel with a BCC to Bob.

It is natural to ask, for example, what Alice has actually learned from Bob's email. Also, do all four people involved in this exchange have common knowledge of the original email by Clare?

To answer such questions I study email exchanges focusing on relevant features that are encountered in most email systems. More specifically, I make the following assumptions:

- each email has a sender, a non-empty set of regular recipients and a (possibly empty) set of blind carbon copy (BCC) recipients. Each recipient receives a copy of the message and is only aware of the regular recipients and not of the BCC recipients (except himself if he is one),

- in the case of a reply to or a forward of a message, the *unaltered* original message is included,
- in a reply or a forward, the list of regular recipients is included but the list of BCC recipients is not,
- in a reply or a forward, one can append new information to the original message one replies to or forwards.

In order to formalize the agents' knowledge resulting from an email exchange I will introduce an appropriate epistemic language and the corresponding semantics. The resulting model of email communication differs from the ones that were studied in other papers in which only limited aspects of emails have been considered. These papers are discussed below.

In my setup the communication is synchronous. This matches the actual situation in the sense that when an email is sent it is in most cases immediately present in the inbox of the recipients. However, this is a simplification since the fact that the email is present in the inbox of the agent does not mean the agent also reads it immediately (or indeed reads it at all). I find that it is natural to clarify email communication in a synchronous setting first before considering alternatives. In Chapter 6 I distinguish two different kinds of knowledge based on the fact that not all emails are read immediately.

5.1.1 Contributions and Plan of this Chapter

To study the relevant features of email communication I will introduce in the next section a carefully chosen language describing emails. I make a distinction between a message, which is sent to a public recipient list, and an email, which consists of a message and a set of BCC recipients. This distinction is relevant because a forward email contains an earlier message, without the list of BCC recipients. I also introduce the notion of a legal state that imposes a natural restriction on the considered sets of emails by stipulating an ordering of the emails. For example, an email needs to precede any forward of it.

To reason about the knowledge of the agents after an email exchange has taken place I introduce in Section 5.3 an appropriate epistemic language. Its semantics takes into account the uncertainty of the recipients of an email about its set of BCC recipients. This semantics allows me to evaluate epistemic formulas in legal states, in particular the formulas that characterize the full knowledge-theoretic effect of an email.

Apart from factual information each email also carries epistemic information. In Section 5.4 I characterize the latter. It allows me to clarify which groups of agents acquire common knowledge as a result of an email and what the resulting information gain for each agent is.

In Section 5.5 I present the main result of the chapter, that clarifies when a group of agents acquires common knowledge of the fact that an email has been sent. This characterization in particular sheds light on the epistemic consequences of BCC. The proof is given in Section 5.6.

Then in Section 5.7 I show that in this framework, BCC cannot be simulated using messages without BCC recipients. Finally, in Section 5.8, I provide a distributed programming perspective of email exchanges. In this view the processes are agents who communicate with emails. I provide an operational semantics of such distributed programs. It allows me to clarify various fine points of email exchanges in the presence of BCC. I then use distributed programs to characterize the notion of a legal state.

5.1.2 Related Work

The study of the epistemic effects of communication in distributed systems originated in the eighties and led to the seminal book [Fagin et al., 1995]. The relevant literature, including [Chandy and Misra, 1985], deals with the communication forms studied within the context of distributed computing, notably asynchronous send.

One of the main issues studied in these frameworks has been the analysis of the conditions that are necessary for acquiring common knowledge. In particular, [Halpern and Moses, 1990] showed that common knowledge cannot be attained in the systems in which the message delivery is not guaranteed. This is exactly the problem that is faced by the generals in the example given in the introduction. More recently this problem was investigated in [Ben-Zvi and Moses, 2010] for synchronous systems with known bounds on message transmission in which processes share a global clock. The authors extended the causality relation of [Lamport, 1978] between messages in distributed systems to synchronous systems with known bounds on message transmission and proved that in such systems a so-called pivotal event is needed in order to obtain common knowledge. This in particular generalizes the previous result of [Chandy and Misra, 1985] concerning acquisition of common knowledge in distributed systems with synchronous communication.

The epistemic effects of other forms of communication were studied in numerous papers. In particular, in [Pacuit and Parikh, 2007] the communicative acts are assumed to consist of an agent j 'reading' an arbitrary propositional formula from another agent i. The idea of the epistemic content of an email is implicitly present in [Parikh and Ramanujam, 2003], where a formal model is proposed that formalizes how communication changes the knowledge of a recipient of the message.

In [van Benthem et al., 2006] a dynamic epistemic logic modeling effects of communication and change is introduced and extensively studied. [Pacuit, 2010] surveys these and related approaches and discusses the used epistemic, dynamic

epistemic and doxastic logics.

In Chapter 3 I have presented a framework that studies the knowledge of agents who communicate via messages. The framework presented there is based on the assumption that there is a fixed set of a finite number of possible messages, and this set is common knowledge among the agents. This is a reasonable assumption in a number of settings, but not in the setting studied in this chapter. In email communication, the number of possible messages is unlimited. Even if one abstracts the message contents and focusses on the lists of recipients and the structure of forwards and replies there is an infinite number of possible combinations. Therefore I need to find a different model for this situation.

Most related to the work here reported is [Apt et al., 2009], which studied knowledge and common knowledge in a set-up in which the agents send and forward propositional formulas in a social network. However, the forward did not include the original message and the BCC feature was absent. Just like in Chapter 3, there it is assumed that the number of messages is finite. In contrast, in the setting of this chapter the forward includes the original message, which results directly in an infinite number of possible messages and emails.

5.2 Preliminaries

5.2.1 Messages

In this section I define the notion of a message. In the next section I introduce emails as simple extensions of the messages. Let a finite set of agents Ag and a finite set of ***notes*** $\mathcal{N}$ be given. The notes represent the contents of the message or an email, just like in Chaper 4.

I will assume that initially each agent a has a set of notes N_a he knows. He does not know which notes belong to the other agents, but he does know the overall set of notes. Furthermore, I assume that an agent can send a message to other agents containing a note only if he holds it initially or has learnt it through a message he received earlier.

Of course in reality emails may contain propositional or epistemic information which affects knowledge of the agents at a deeper level than modeled here by means of abstract notes. To reason about notes containing such information one could add on the top of my framework an appropriate logic. If every note n contains some formula φ_n, then one could just add the implications $n \rightarrow \varphi_n$ to this logic to ensure that every agent who knows the note n also knows the formula φ_n.

This minimal set-up precludes that the agents can use messages to implement some protocol that was agreed in advance, such as that sending two specific notes by an agent would reveal that he has some specific knowledge. It allows me to focus instead on the epistemic information caused *directly* by the structure of the

messages and emails.

I inductively define a ***message*** as a construct of one of the following forms:

- $m := s(a, n, G)$; the message containing note n, sent by a to the group G,
- $m := f(a, n.m', G)$; the forwarding by agent a of the message m' with added note n, sent to the group G.

So the agents can send a message with a note or forward a message with a new note appended, where the latter covers the possibility of a reply or a reply-all. Appending such a new note to a forwarded message is a natural feature present in most email systems. To allow for the possibility of sending a forward without appending a new note, I assume there exists a note **true** that is held by all agents and identify $\mathbf{true}.m$ with m.

Just like in Chapter 3, I use s_m and r_m for the sender and the group of recipients of a message m, respectively. So for the above messages m I have $s_m = a$ and $r_m = G$. I do allow that $s_m \subseteq r_m$, i.e., that one sends a message to oneself.

Special forms of the forward messages can be used to model reply messages. Given $f(a, n.m, G)$ with $a \in r_m$, using $G = \{s_m\}$ results in the customary *reply* message and using $G = \{s_m\} \cup r_m$ results in the *reply-all* message. In the customary email systems there is syntactic difference between a forward and a reply to these two groups of agents, but the effect of both messages is exactly the same, so I ignore this difference. In the examples I write $s(a, n, b)$ instead of $s(a, n, \{b\})$, etc.

5.2.2 Emails

An interesting feature of most email systems is that of the blind carbon copy (BCC). I will now include this in my framework.

In the previous subsection I defined messages that have a sender and a group of recipients. Now I define the notion of an email which allows the additional possibility of sending a BCC of a message. Formally, by an ***email*** I mean a construct of the form m_B, where m is a message and $B \subseteq Ag$ is a possibly empty set of BCC recipients. Given a message m I call each email of the form m_B a ***full version*** of m, and say that it is ***based on*** m.

An email m_B is delivered to the regular recipients, i.e., to the set r_m, and to the set B of BCC recipients. Each of them receives the message m. Only the sender of m_B, i.e., the agent s_m, knows the set B. Each agent $a \in B$ only knows that the set B contains at least him.

Since the set of BCC recipients is 'secret', it does not appear in a forward. That is, a forward of an email m_B with added note n is a message $f(a, n.m, G)$ or an email $f(a, n.m, G)_C$, in which B is not mentioned. This is consistent with the way BCC is handled in most email systems, such as `gmail` or email systems

based on the `postfix` mail server. However, this forward may be sent not only by a sender or a regular recipient of m_B, but also by a BCC recipient. Clearly, the fact that an agent was a BCC recipient of an email is revealed at the moment he forwards the message.

A natural question arises: what if someone is both a regular recipient and a BCC recipient of an email? In this case, no one (not even this BCC recipient himself) would ever notice that this recipient was also a BCC recipient since everyone can explain his knowledge of the message by the fact that he was a regular recipient. Only the sender of the message would know that this agent was also a BCC recipient. This fact does not have any noticeable consequences and hence I will assume that for every email m_B it holds that $(\{s_m\} \cup r_m) \cap B = \emptyset$.

5.2.1. Example. Using the newly introduced language I can formalize the story from Example 5.1.1 as follows, where I abbreviate Alice to a, etc.:

- Alice and Daniel got an email from Clare:

$$e_0 := m_\emptyset, \text{ where } m := s(c, n, \{a, d\}),$$

- Alice forwarded it to Bob:

$$e_1 := m'_\emptyset, \text{ where } m' := f(a, m, b),$$

- Bob forwarded Alice's email to Clare and Daniel with a BCC to Alice:

$$e_2 := m''_{\{a\}}, \text{ where } m'' := f(b, m', \{c, d\}),$$

- Alice forwarded the last email to Clare and Daniel with a BCC to Bob:

$$e_3 := f(a, m'', \{c, d\})_{\{b\}}.$$

5.2.3 Legal States

My goal is to analyze a collection of sent emails in order to find out what knowledge the agents acquired from them. In this section I will state some properties that I will assume such a collection of emails has in order to be realistic.

First of all, I shall assume that for each message m there is at most one full version of m, i.e., an email of the form m_B. The rationale behind this decision is that a sender of m_B and $m_{B'}$ might just as well send a single email $m_{B \cup B'}$. This assumption can be summarized as a statement that the agents do not have 'second thoughts' about the recipients of their emails. It also simplifies subsequent considerations.

I have decided not to impose a total ordering on the emails in the model, for example by giving each email a time stamp. This makes the model a lot simpler.

Also, many interesting questions can be answered without imposing such a total ordering. For example, I can investigate the existence of common knowledge in a group of agents after an email exchange perfectly well without knowing the exact order of the emails that were sent.

However, I will impose a partial ordering on the sets of emails. This is useful because I need to make sure that the agents only send information they actually know. Moreover, a forward can only be sent after the original email was sent. I will introduce the minimal partial ordering that takes care of these issues.

First, I define by structural induction the ***factual information*** $FI(m)$ contained in a message m as follows:

$$\begin{aligned} FI(s(a,n,G)) &:= \{n\}, \\ FI(f(a,n.m,G)) &:= FI(m) \cup \{n\}. \end{aligned}$$

Informally, the factual information is the set of notes which occur somewhere in the message, including those occurring in forwarded messages.

I will represent an email exchange as a ***state*** $s = (E, N)$. It is a tuple consisting of a finite set E of emails that were sent and a sequence $N = (N_1, \ldots, N_n)$ of sets of notes for all agents. The idea of these sets is that each agent a initially holds the notes in N_a. I use E_s and N_s to denote the corresponding elements of a state s, and $N_1, ..., N_n$ to denote the elements of N.

I say that a state $s = (E, N)$ is ***legal*** if a strict partial ordering (in short, an spo) $\prec$ on E exists that satisfies the following conditions:

L.1: for each email $f(a, n.m, G)_B \in E$ an email $m_C \in E$ exists such that $m_C \prec f(a, n.m, G)_B$ and $a \in \{s_m\} \cup r_m \cup C$,

L.2: for each email $s(a, n, G)_B \in E$, where $n \notin N_a$, an email $m_C \in E$ exists such that $m_C \prec s(a, n, G)_B$, $a \in r_m \cup C$ and $n \in FI(m)$,

L.3: for each email $f(a, n.m', G)_B \in E$, where $n \notin N_a$, an email $m_C \in E$ exists such that $m_C \prec f(a, n.m', G)_B$, $a \in r_m \cup C$ and $n \in FI(m)$.

Condition L.1 states that the agents can only forward messages they previously received. Conditions L.2 and L.3 state that if an agent sends a note that he did not initially hold, then he must have learnt it by means of an earlier email.

So a state is legal if its emails can be partially ordered in such a way that every forward is preceded by its original message, and for every note sent in an email there is an explanation how the sender of the email learnt this note. As every partial ordering can be extended to a linear ordering, the emails of a legal state can be ordered in such a way that each agent has a linear ordering on its emails. However, such a linear ordering does not need to be unique. For example, the emails $s(a, n, b)_\emptyset$ and $s(a, n, c)_\emptyset$ can always be ordered in both ways.

Moreover, a strict partial ordering that ensures that a state is legal does not need to be unique either and incompatible minimal partial orderings can exist.

Here is an example. Suppose that $n \in N_a \backslash N_b$ and $b \in G_1 \cap G_2$, and consider the set of messages $\{s(a,n,G_1),\ s(a,n,G_2),\ s(b,n,c)\}$. The resulting state (we identify here each message m with the email $m_\emptyset$) is legal. There are two minimal spos that can be used to establish this, $s(a,n,G_1) \prec s(b,n,c)$ and $s(a,n,G_2) \prec s(b,n,c)$. So it cannot be assumed that any specific message sent by agent a has to precede the message sent by agent b, though it must be so that at least one of them does.

This shows that the causal relation between emails essentially differs from the causal relation between messages in distributed systems, as studied in [Lamport, 1978]. Furthermore, the assumption that communication is synchronous does not result in a unique spo on the considered emails.

Because of the lack of a unique spo on the emails it is tempting to use an alternative definition that stipulates that each email is 'justified' by a set of emails. For instance, in the above example the message $s(b,n,c)$ is justified by the set $\{s(a,n,G_1),\ s(a,n,G_2)\}$. Unfortunately, because of the fact that it is possible to append notes to forwarded messages, this is not a valid alternative. Indeed, consider the following set of messages

$$\{s(1,p,2),\ s(1,q,3),\ s(1,r,4),\\ f(2,r.s(1,p,2),3),\ f(3,p.s(1,q,3),4),\ f(4,q.s(1,r,4),2)\},$$

and assume that $p,q,r \in N_1$ and $p,q,r \notin N_2 \cup N_3 \cup N_4$. Then each message has a justification. For example the message $f(2,r.s(1,p,2),3)$ can be justified by the set $\{s(1,p,2),\ f(4,q.s(1,r,4),2)\}$. Indeed, the first message justifies the '$s(1,p,2)$' component and the second one justifies the 'r' component. However, it is easy to see that this is not a legal state: each of the notes appended to the forwards can only be known by the sender after one of the other forwards has been received. Therefore, none of the forwards can be the first forward.

5.3 Epistemic Language and its Semantics

In order to reason about the knowledge of the agents after an email exchange has taken place I introduce the language $\mathcal{L}_{EE}$ of email exchanges as follows:

$$\varphi := m \mid i \blacktriangleleft m \mid \neg\varphi \mid \varphi \wedge \varphi \mid C_G\varphi$$

Here m denotes a message. The formula m expresses the fact that m has been sent in the past, with some unknown group of BCC recipients. The formula $i \blacktriangleleft m$ expresses the fact that agent i was involved in a full version of the message m, i.e., he was either the sender, a recipient or a BCC recipient. The formula $C_G\varphi$ denotes common knowledge of the formula φ in the group G.

I use the usual abbreviations $\vee$, $\rightarrow$ and $\leftrightarrow$ and use $K_i\varphi$ as an abbreviation of $C_{\{i\}}\varphi$. The fact that an email with a certain set of BCC recipients was sent can be expressed in this language with the following abbreviation:

$$m_B := m \wedge \bigwedge_{i \in \{s_m\} \cup r_m \cup B} i \blacktriangleleft m \wedge \bigwedge_{i \notin \{s_m\} \cup r_m \cup B} \neg i \blacktriangleleft m$$

Note that this formula expresses the fact that the message m was sent with exactly the group B as BCC recipients, which captures precisely the intended meaning of m_B.

I will now provide a semantics for this language interpreted on legal states, inspired by the perspective of epistemic logic and the history-based approaches of [Pacuit and Parikh, 2007] and [Parikh and Ramanujam, 2003]. For every agent a I define an indistinguishability relation $\sim_a$, where I intend $s \sim_a s'$ to mean that agent a cannot distinguish between the states s and s'. I first define this relation on the level of emails as follows:

$$m_B \sim_a m'_{B'}$$

iff one of the following contingencies holds:

(i) $s_m = a$, $m = m'$ and $B = B'$,

(ii) $a \in r_m \setminus \{s_m\}$ and $m = m'$,

(iii) $a \in B \cap B'$ and $m = m'$.

Recall that I assume that senders and regular recipients are not BCC recipients, so conditions (i) - (iii) are mutually exclusive. Condition (i) states that the sender of an email confuses it only with the email itself. In turn, condition (ii) states that each regular recipient of an email who is not a sender confuses it with any email with the same message but possibly sent to a different BCC group. Condition (iii) states that each BCC recipient of an email confuses it with any email with the same message but sent to a possibly different BCC group of which he is also a member. Finally, condition (iv) states that if a is no sender, regular recipient or BCC recipient of m or m' then he confuses them. It will become clear that in this case the question of whether a confuses these messages is irrelevant for the proceedings. Since a has nothing to do with these messages in this case, it is not important to know whether he can distinguish them. However, the fact that a confuses the two messages matches the intuition that a knows nothing about these messages.

5.3.1. Example. Consider the emails $e := s(a, n, b)_\emptyset$ and $e' := s(a, n, b)_{\{c\}}$. Then $e \not\sim_a e'$, $e \sim_b e'$ and $e \not\sim_c e'$. Intuitively, agent b cannot distinguish between these two emails because he cannot see whether c is a BCC recipient. In contrast, agents a and c can distinguish between these two emails.

Next, I extend the indistinguishability relation to legal states by defining

$$(E, N) \sim_a (E', N')$$

iff all of the following hold:

- $N_a = N'_a$,
- for every $m_B \in E$ such that $a \in \{s_m\} \cup r_m \cup B$ there is an $m_{B'} \in E'$ such that $m_B \sim_a m_{B'}$,
- for every $m_{B'} \in E'$ such that $a \in \{s_m\} \cup r_m \cup B'$ there is an $m_B \in E$ such that $m_B \sim_a m_{B'}$.

So two states cannot be distinguished by an agent if they agree on his notes and their email sets look the same to him. Since I assume that the agents do not know anything about the other notes, I do not refer to the sets of notes of the other agents. Note that $\sim_a$ is an equivalence relation.

5.3.2. Example. Consider the legal states s_1 and s_2 which are identical apart from their sets of emails:

$$\begin{array}{lcl} E_{s_1} & := & \{s(a,n,b)_\emptyset, f(b,s(a,n,b),d)_\emptyset\}, \\ E_{s_2} & := & \{s(a,n,b)_{\{c\}}, f(b,s(a,n,b),d)_\emptyset, f(c,s(a,n,b),d)_\emptyset\}. \end{array}$$

I assume here that $n \in N_a$ and that in each state the emails are ordered by the textual ordering. So in the first state agent a sends a message with note n to agent b and then b forwards this message to agent d. Furthermore, in the second state agent a sends the same message but with a BCC to agent c, and then both agent b and agent c forward the message to agent d.

From the above definition it follows that $s_1 \not\sim_a s_2$, $s_1 \sim_b s_2$, $s_1 \not\sim_c s_2$ and $s_1 \not\sim_d s_2$. For example, $s_1 \not\sim_a s_2$ holds because, as noticed above, $s(a,n,b)_\emptyset \not\sim_a s(a,n,b)_{\{c\}}$. Intuitively, in state s_1 agent a is aware that he sent a BCC to nobody, while in state s_2 he is aware that he sent a BCC to agent c. In turn, in both states s_1 and s_2 agent b is aware that he received the message $s(a,b,n)$ and that he forwarded the email $f(b,s(a,n,b),d)_\emptyset$. Intuitively, in state s_2 agent b does not notice the BCC of the message $s(a,b,n)$ and is not aware of the email $f(c,s(a,b,n),d)_\emptyset$.

In order to express common knowledge, I define for a group of agents G the relation $\sim_G$ as the reflexive transitive closure of $\bigcup_{a\in G} \sim_a$. Then I define the truth of a formula from our language in a state inductively as follows, where $s = (E, N)$:

$$\begin{array}{lll} s \models m & \text{iff} & \exists B : m_B \in E \\ s \models a \blacktriangleleft m & \text{iff} & \exists B : m_B \in E \text{ and } a \in \{s_m\} \cup r_m \cup B \\ s \models \neg\varphi & \text{iff} & s \not\models \varphi \\ s \models \varphi \wedge \psi & \text{iff} & s \models \varphi \text{ and } s \models \psi \\ s \models C_G\varphi & \text{iff} & s' \models \varphi \text{ for every legal state } s' \text{ such that } s \sim_G s' \end{array}$$

I say that φ is ***valid*** (and often just write 'φ' instead of 'φ is valid') if for all legal states s, $s \models \varphi$.

Even though this definition does not specify the form of communication, one can deduce from the definition of the relation $\sim$ that the communication is synchronous, that is, that each email is simultaneously received by all the recipients. Note also that the condition of the form $m_B \in E$ present in the second clause implies that for every email m_B the following equivalence is valid for all $a, b \in \{s_m\} \cup r_m \cup B$:

$$a \blacktriangleleft m \leftrightarrow b \blacktriangleleft m.$$

This means that in every legal state (E, N) either all recipients of the email m_B received it (when $m_B \in E$) or none (when $m_B \notin E$).

However, it should be noted that the agents do not have a common 'clock' using which they could deduce how many messages have been sent by other agents between two consecutive messages they have received. Furthermore, the agents do not have a local 'clock' using which they could count how many messages they sent or received.

When I say that a message m' is ***mentioned in*** or ***a part of*** another message m I mean that m is m' itself, or a forward of m', or a forward of a forward of m', and so on.

The following lemma clarifies when specific formulas are valid. In the sequel I shall use these observations implicitly.

5.3.3. Lemma.

(i) $m \rightarrow m'$ *is valid iff* m' *is part of the message* m.

(ii) $m \rightarrow a \blacktriangleleft m'$ *is valid iff either* $m \rightarrow m'$ *is valid and* $a \in \{s_{m'}\} \cup r_{m'}$ *or for some note* n *and group* G, $f(a, n.m', G)$ *is part of the message* m.

The second item states that $m \rightarrow a \blacktriangleleft m'$ is valid either if a is a sender or a receiver of m' (in that case actually $m \rightarrow a \blacktriangleleft m'$ is valid) or if m shows that a forwarded the message m'. The latter is also possible if a was a BCC receiver of m'. The claimed equivalence holds thanks to condition L.1.

5.3.4. Example. To illustrate the definition of truth, let me return to Example 5.3.2. In state s_2 agent b does not know that agent c received the message $s(a, n, b)$ since he cannot distinguish s_2 from the state s_1 in which agent c did not receive this message. So $s_2 \models \neg K_b c \blacktriangleleft s(a, n, b)$ holds.

On the other hand, in every legal state s_3 such that $s_2 \sim_d s_3$ both an email $f(c, s(a, n, b), d)_C$ and a 'justifying' email $s(a, n, b)_B$ have to exist such that $s(a, n, b)_B \prec f(c, s(a, n, b), d)_C$ and $c \in B$, where $\prec$ is an spo such that the emails of s_3 satisfy conditions L.1-L.3 w.r.t. $\prec$. Consequently $s_3 \models c \blacktriangleleft s(a, b, n)$, so $s_2 \models K_d c \blacktriangleleft s(a, n, b)$ holds, so by sending the forward agent c revealed himself to d as a BCC recipient.

I leave it to the reader to check that both $s_2 \models C_{\{c,d\}} c \blacktriangleleft s(a, n, b)$ and $s_2 \models \neg C_{\{b,d\}} c \blacktriangleleft s(a, n, b)$ hold. In words, agents c and d have common knowledge

that agent c was involved in a full version of the message $s(a, n, b)$, while the agents b and d do not.

5.4 Epistemic Contents of Emails

In Subsection 5.2.3 I defined the factual information contained in a message. Using epistemic formulas I can also define the ***epistemic information*** contained in a message or an email. First, I define it for messages as follows:

$$\begin{aligned} EI(s(a,n,G)) &:= C_{\{a\}\cup G} s(a,n,G), \\ EI(f(a,n.m,G)) &:= C_{\{a\}\cup G}(f(a,n.m,G) \wedge EI(m)). \end{aligned}$$

So the epistemic information contained in a message is the fact that the sender and receivers acquire common knowledge of the message. In the case of a forward the epistemic information contained in the original message also becomes common knowledge. This results in nested common knowledge. In general, iterated forwards can lead to arbitrary nesting of the common knowledge operator, each time involving a different group of agents.

The definition of the epistemic information contained in an *email* additionally needs to capture the information about the agents who are on the BCC list of an email. I define:

$$EI(m_B) := EI(m) \wedge \bigwedge_{a \in B} C_{\{s_m\}\cup\{a\}}(EI(m) \wedge a \blacktriangleleft m) \wedge K_{s_m} m_B.$$

So $EI(m_B)$ states that

- the epistemic information contained in the message m holds,
- the sender of the message and each separate agent on the BCC list have common knowledge of this epistemic information and of the fact that this agent received the message,
- the sender knows the precise set of BCC recipients.

The following result shows that indeed the epistemic information in a message or an email holds in a state if and only if the message or email was sent.

5.4.1. Theorem. *The following equivalences are valid:*

(i) $m \leftrightarrow EI(m)$,

(ii) $m_B \leftrightarrow EI(m_B)$.

PROOF. Each relation $\sim_a$ on the level of states is an equivalence relation, so for all formulas φ and $G \subseteq Ag$, the implication $C_G\varphi \to \varphi$, and hence in particular $EI(m) \to m$ and $EI(m_B) \to m_B$, is valid.

(i) To prove the validity of $m \to EI(m)$, take some message m. Let $A = \{s_m\} \cup r_m$. Consider an arbitrary legal state s and assume that $s \models m$. Suppose $s \sim_A s'$ for some legal state s'. Then there is a path $s = s_0 \sim_{a_1} s_1 \sim_{a_2} \ldots \sim_{a_l} s_l = s'$ from s to s', where $a_1, \ldots, a_l \in A$.

For every $k \in \{1, \ldots, l\}$ suppose $s_k = (E_k, N_k)$. Then for every $k \in \{1, \ldots, l\}$, $s_{k-1} \models m$ implies that for some B, $m_B \in E_{k-1}$. Now, since $a_k \in \{s_m\} \cup r_m$, by the clauses (i) and (ii) of the definition of the $\sim_{i_k}$ relation on the emails for some group B' I have $m_{B'} \in E_k$, which implies $s_k \models m$. Since $s \models m$, an inductive argument shows that $s' \models m$. This proves that $s \models C_A m$. So I established the validity of the implication

$$m \to C_A m,$$

and in particular of $s(i, l, G) \to EI(s(i, l, G))$.

For the forward messages I proceed by induction on the structure of the messages. The base case is given by the implication $s(i, l, G) \to EI(s(i, l, G))$. Consider the message $f(a, n.m, G)$. The implication $f(a, n.m, G) \to m$ is valid, so by the induction hypothesis the implication $f(a, n.m, G) \to EI(m)$ is valid. Since I showed already that the implication

$$f(a, n.m, G) \to C_{\{a\} \cup G} f(a, n.m, G)$$

is valid, I conclude that the implication

$$f(a, n.m, G) \to C_{\{a\} \cup G}(f(a, n.m, G) \wedge EI(m))$$

is also valid.

(ii) I already established the validity of $m \to EI(m)$. Then by the definition of m_B the implication $m_B \to EI(m)$ is also valid.

Let $a \in B$. Consider an arbitrary legal state s and assume that $s \models m_B$. Suppose $s \sim_{\{s_m\} \cup \{a\}} s'$ for some legal state s'. Then there is a path $s = s_0 \sim_{a_1} s_1 \sim_{a_2} \ldots \sim_{a_l} s_l = s'$ from s to s', where $a_1, \ldots, a_l \in \{s_m\} \cup \{a\}$ and $l \geq 0$.

For every $k \in \{1, \ldots, l\}$ suppose $s_k = (E_k, N_k)$. Then for every $k \in \{1, \ldots, l\}$, $s_{k-1} \models m_B$ implies that $m_B \in E_{k-1}$ and then by the definition of $\sim_k$, $m_{B'} \in E_k$ for some B' such that $a \in B'$. This means that $s_k \models a \blacktriangleleft m$ and $s_k \models m$ which implies by (i) that $s_k \models EI(m)$. Since $s \models m_B$ an inductive argument then shows that $s' \models EI(m) \wedge a \blacktriangleleft m$. So $s \models C_{\{s_m\} \cup \{a\}}(EI(m) \wedge a \blacktriangleleft m)$.

Finally, suppose that $s \sim_b s'$, where $\{s_m\} = \{b\}$, and $s \models m_B$. By the definition of the $\sim_b$ relation on the level of states $m_B \in E_{s'}$ so $s' \models m_B$. This proves $s \models K_{s_m} m_B$.

I conclude that the implication $m_B \to EI(m_B)$ is valid. Trivially, $EI(m_B) \to m_B$ is also valid. □

Using the above theorem it can be determined 'who knows what' after an email exchange E (taken from a legal state (E, N)) took place. The problem boils down to computing $\bigwedge_{e\in E} EI(e)$. When one is interested in a specific fact, for example whether after an email exchange E took place agent i knows a formula ψ, one simply needs to establish the validity of the implication $\bigwedge_{e\in E} EI(e) \to C_a\psi$.

Using the epistemic information contained in an email I can define the ***information gain*** of an agent resulting from sending or receiving of an email as follows. Suppose $a \in \{s_m\} \cup r_m \cup B$. Then

$$IG(m_B, a) := \begin{cases} EI(m_B) & \text{if } s_m = a \\ EI(m) & \text{if } a \in r_m \\ C_{\{s_m\}\cup\{a\}}(EI(m) \wedge a \blacktriangleleft m) & \text{if } a \in B \end{cases}$$

Then the following result is a simple consequence of Theorem 5.4.1.

5.4.2. Corollary. *Take a legal state $s = (E, N)$ and an email $m_B \in E$. Then for every agent $a \in \{s_m\} \cup r_m \cup B$,*

$$s \models K_a IG(m_B, a).$$

Proof. It follows immediately from Theorem 5.4.1 that for any $a \in \{s_m\} \cup r_m \cup B$, $s \models IG(m_B, a)$. A closer inspection of the form of $IG(m_B, a)$ reveals that for any such a, $IG(m_B, a) \to K_a IG(m_B, a)$. So $s \models K_a IG(m_B, a)$. □

5.4.3. Example. Using the notion of an information gain I can answer the first question posed in Example 5.1.1, namely what Alice learned from Bob's email. First I recall the messages and emails defined there:

$$\begin{array}{lcll} m & := & s(c, n, \{a, d\}), & \\ e_1 & := & m'_{\emptyset}, & \text{where } m' := f(a, m, b), \\ e_2 & := & m''_{\{a\}}, & \text{where } m'' := f(b, m', \{c, d\}). \end{array}$$

By definition,

$$\begin{array}{lcl} EI(m) & = & C_{\{a,c,d\}} m, \\ EI(m') & = & C_{\{a,b\}}(m' \wedge EI(m)), \\ EI(m'') & = & C_{\{b,c,d\}}(m'' \wedge EI(m')), \\ IG(e_2, a) & = & C_{\{a,b\}}(EI(m'') \wedge b \blacktriangleleft m''). \end{array}$$

This should be contrasted with the information Alice had after she sent the email e_1, which was $EI(m')$.

5.5 Common Knowledge

I will now clarify when a group of agents acquires common knowledge of the formula expressing that an email was sent. This shows how my framework can be used to investigate epistemic consequences of email exchanges.

Given a set of emails E and a group of agents A, let the group of emails ***shared by the group*** A be defined as

$$E_A := \{m_B \in E \mid A \subseteq \{s_m\} \cup r_m \text{ or } \exists b \in B : (A \subseteq \{s_m\} \cup \{b\})\}.$$

Note that when $|A| \geq 3$, then $e \in E_A$ iff $A \subseteq \{s_m\} \cup r_m$. When $|A| = 2$, then $e \in E_A$ also when $\exists j \in B : A = \{s_m\} \cup \{j\}$, and when $|A| = 1$, then $e \in E_A$ also when $A = \{s_m\}$ or $\exists j \in B : A = \{j\}$.

The following theorem uses this definition to provide a simple way of testing whether a message or an email is common knowledge in a group of agents.

5.5.1. Theorem. ***Main Theorem*** *Consider a legal state $s = (E, N)$ and a group of agents A.*

(*i*) *$s \models C_A m$ iff there is $m'_B \in E_A$ such that $m' \to m$ is valid.*

(*ii*) *Suppose that $|A| \geq 3$. Then $s \models C_A m_B$ iff the following hold:*

C1 *$\{s_m\} \cup r_m \cup B = Ag$,*

C2 *for each $b \in B$ there is $m'_{B'} \in E_A$ such that $m' \to b \blacktriangleleft m$ is valid,*

C3 *there is $m'_{B'} \in E_A$ such that $m' \to m$ is valid.*

Part (*i*) shows that when I limit my attention to messages, then things are as expected: a group of agents acquires common knowledge of a message m iff they receive an email that mentions m. If I limit my presentation to emails with the empty BCC sets I get as a direct corollary the counterpart of this result for a simplified framework with messages only.

To understand part (*ii*) note that it states that $s \models C_A m_B$ iff

- the email m_B involves all agents (recall that Ag is the set of all agents),
- for every agent b that is on the BCC list of m_B there is an email shared by the group A that proves that b was involved in message m, i.e., that b forwarded the message m,
- there is an email shared by the group A that proves the existence of the message m.

The first of the above three items is striking and shows that common knowledge of an email is rare. **C3** is just the condition used in part (i). So an email m_B such that $A \subseteq \{s_m\} \cup r_m$ does ensure that the group of agents A acquires common knowledge of m. However, the group A can never know what was the set of the BCC recipients of m_B *unless* it was the set $Ag \setminus (\{s_m\} \cup r_m)$ *and* there is a proof for this fact in the form of the 'disclosing emails' from all members of B.

Having in mind that the usual purpose of the BCC is just to inform its recipients of a certain message (that they are supposed to 'keep for themselves'), I conclude that the presence of the BCC feature essentially precludes that a group of agents can acquire common knowledge of an email. Informally, the fact that the BCC feature creates 'secret information' has as a consequence that common knowledge of an email is only possible if this secret information is completely disclosed to the group in question. Moreover, the message has to be sent to all agents since otherwise the agents might consider the possibility that the other agents also received a BCC.

Note that using the notion of the information gain introduced in the previous section I can determine for each agent in a group A what he learned from a message m or an email m_B. In some circumstances, like when $m = s(i, l, G)$ and $A \subseteq G \cup \{i\}$, this information gain can imply $C_A m$. However, the definition of $EI(m_B)$ implies that the information gain can imply $C_A m_B$ only in the obvious case when $A = \{s_m\}$.

Finally, the above result crucially depends on the fact that the notes are uninterpreted. If one allows emails that contain propositional formulas of the language $\mathcal{L}_{EE}$ from Section 5.3 augmented by the notes, then an agent could communicate to a group A the fact that he sent an email m_B (with a precise set of the BCC recipients). Then m_B would become common knowledge in the group A.

As an aside let me mention that there is a corresponding result for the case when $|A| < 3$, as well. However, it involves a tedious case analysis concerning the possible relations between $A, \{s_m\}, r_m$ and B, so I do not present it here.

5.5.2. Example. I can use the above result to answer the second question posed in Example 5.1.1. Let s be the state whose set of emails consist of the considered four emails, so

$$\begin{array}{ll} e_0 := m_\emptyset, & \text{where } m := s(c, n, \{a, d\}), \\ e_1 := m'_\emptyset, & \text{where } m' := f(a, m, b), \\ e_2 := m''_{\{a\}}, & \text{where } m'' := f(b, m', \{c, d\}), \\ e_3 := f(a, m'', \{c, d\})_{\{b\}}. & \end{array}$$

Alice's set of notes in s consists of n while the sets of notes of Bob, Clare and Daniel are empty. Note that s is legal. Then it holds that

$$s \not\models C_{\{a,b,c,d\}} s(c, n, \{a, d\}).$$

The reason is that

$$E_{\{a,b,c,d\}} = \emptyset.$$

Indeed, for no $m^* \in \{m, m', m'', f(a, m'', \{c, d\})\}$ I have

$$\{a, b, c, d\} \subseteq S(m^*) \cup R(m^*)$$

and for no $m^*_B \in \{e_0, e_1, e_2, e_3\}$ I have some $x \in B$ such that

$$\{a, b, c, d\} \subseteq S(m^*) \cup \{x\}.$$

So there are no messages that ensure common knowledge in the group $\{a, b, c, d\}$. So even though there have been three forwards of the original message, it is not common knowledge.

Clearly, if the original message $s(c, n, \{a, d\})$ is not common knowledge then its forward $f(a, m, b)$ is not common knowledge either. Another way to derive this is directly from the Main Theorem. Namely, I have

$$s \not\models C_{\{a,b,c,d\}} f(b, m', \{c, d\})_{\{a\}}.$$

The reason is that condition **C2** does not hold since no email shared by $\{a, b, c, d\}$ exists that proves that Alice received m''. In contrast,

$$s \models C_{\{a,c,d\}} f(b, m', \{c, d\})_{\{a\}}$$

does hold, since the email e_3 is shared by $\{a, c, d\}$. Furthermore, if Alice had included Daniel in the forward instead of sending him a BCC, and had used the forward $f(a, m'', \{b, c, d\})_\emptyset$, then condition **C2** would hold and I could conclude for this modified state s' that

$$s' \models C_{\{a,b,c,d\}} f(b, m', \{c, d\})_{\{a\}}.$$

5.6 Proof of the Main Theorem

I first establish a number of auxiliary lemmas. I shall use a new strict partial ordering on emails. I define

$$m_B < m'_{B'} \text{ iff } m \neq m' \text{ and } m' \rightarrow m.$$

Note that by Lemma 5.3.3 $m \neq m'$ and $m' \rightarrow m$ precisely if m' is a forward, or a forward of a forward, etc, of m. Then for two emails m_B and $m'_{B'}$ from a legal state s that satisfies conditions L.1-L.3 w.r.t. an spo $\prec$, $m_B < m'_{B'}$ implies $m_B \prec m'_{B'}$ on the account of condition L.1. However, the converse does not need to hold since $m_B \prec m'_{B'}$ can hold on the account of L.2 or L.3. Furthermore, note that the $<$-maximal elements of E are precisely the emails in E that are not forwarded.

Given a set of emails E and $E' \subseteq E$, I define the ***downward closure*** of E' as

$$E'_{\leq} := E' \cup \{e \in E \mid \exists e' \in E' : e < e'\}.$$

The set of emails E on which the downward closure of E' depends will always be clear from the context.

Next, I introduce two operations on states. Assume a state (E, N) and an email $m_B \in E$.

I define the state

$$s \setminus m_B := (E \setminus \{m_B\}, N'),$$

with

$$N'_a := \begin{cases} N_a \cup FI(m) & \text{if } a \in r_m \cup B \\ N_a & \text{otherwise} \end{cases}$$

Intuitively, $s \setminus m_B$ is the result of removing the email m_B from the state s, followed by augmenting the sets of notes of its recipients in such a way that they initially already had the notes they would have acquired from m_B. Note that $s \setminus m_B$ is a legal state if m_B is an $<$-maximal element of E.

Next, given $C \subseteq B$ I define the state

$$s[m_{B \mapsto C}] := (E \setminus \{m_B\} \cup \{m_C\}, N'),$$

with

$$N'_a := \begin{cases} N_a \cup FI(m) & \text{if } a \in B \setminus C \\ N_a & \text{otherwise} \end{cases}$$

Intuitively, $s[m_{B \mapsto C}]$ is the result of shrinking the set of BCC recipients of m from B to C, followed by an appropriate augmenting of the sets of notes of the agents that no longer receive m.

Note that $s[m_{B \mapsto C}]$ is a legal state if there is no forward of m by an agent $a \in B \setminus C$, i.e., no email of the form $f(a, n.m, G)_D$ exists in E such that $a \in B \setminus C$.

I shall need the following lemma that clarifies the importance of the set E_A of emails.

5.6.1. Lemma. *Consider a legal state $s = (E, N)$ and a group of agents A. Then for some N' the state $s' := ((E_A)_{\leq}, N')$ is legal and $s \sim_A s'$.*

Proof. I prove that for all $<$-maximal emails $m_B \in E$ such that $m_B \notin E_A$ (so neither $A \subseteq \{s_m\} \cup r_m$ nor $\exists a \in B : (A \subseteq \{s_m\} \cup \{a\})$) I have $s \sim_A s \setminus m_B$. Iterating this process I get the desired conclusion.

Suppose m_B is a $<$-maximal email in E such that $m_B \notin E_A$. Take some $a \in A \setminus (\{s_m\} \cup r_m)$. Suppose first $a \notin B$. Then $s \sim_a s \setminus m_B$ so $s \sim_A s \setminus m_B$.

Suppose now $a \in B$. Define

$$s_1 := s[m_{B \mapsto \{a\}}].$$

Then s_1 is a legal state and $s \sim_a s_1$. Next, define

$$s_2 := s[m_{B \mapsto \emptyset}].$$

Now take some $b \in A \setminus (\{s_m\} \cup \{a\})$. Then $s_1 \sim_b s_2 \sim_a s \setminus m_B$ so $s \sim_A s \setminus m_B$. Note that both s_1 and s_2 are legal states since m_B is $<$-maximal. □

Using the above lemma I now establish two auxiliary results concerning common knowledge of the formula $a \blacktriangleleft m$ or of its negation.

5.6.2. Lemma.

(i) $s \models C_A a \blacktriangleleft m$ *iff* $\exists m'_B \in E_A : (m' \to a \blacktriangleleft m)$
or $(A \subseteq \{s_m\} \cup \{a\}$ *and* $\exists m_B \in E_A : (a \in B))$.

(ii) $s \models C_A \neg a \blacktriangleleft m$ *iff* $s \models \neg a \blacktriangleleft m$ *and* $(A \subseteq \{s_m\} \cup \{a\}$ *or* $s \models C_A \neg m)$.

To illustrate various alternatives listed in (i) note that each of the following emails in E ensures that $s \models K_b a \blacktriangleleft m$, where in each case m is the corresponding send message:

$$s(a, n, G)_{\{b\}}, f(c, q.s(a, n, G), H)_{\{b\}},$$
$$s(c, n, a)_{\{b\}},\ f(a, q.s(c, n, G), H)_{\{b\}},\ s(b, n, G)_{\{a\}}.$$

The first four of these emails imply $s \models K_b a \blacktriangleleft m$ by the first clause of (i), the last one by the second clause.

Proof. (i) $(\Rightarrow)$ Suppose $s \models C_A a \blacktriangleleft m$. Take the legal state s' constructed in Lemma 5.6.1. Then $s \sim_A s'$, so $s' \models a \blacktriangleleft m$. Hence for some group B I have $m_B \in (E_A)_{\leq}$ and $a \in \{s_m\} \cup r_m \cup B$. Three cases arise.

Case 1: $a \in \{s_m\} \cup r_m$.
Then $m \to a \blacktriangleleft m$. So if $m_B \in E_A$, then the claim holds. Otherwise some email $m'_{B'} \in E_A$ exists such that $m_B < m'_{B'}$. Consequently $m' \to m$ and hence $m' \to a \blacktriangleleft m$. So the claim holds as well.

Case 2: $a \notin \{s_m\} \cup r_m$ and $A \subseteq \{s_m\} \cup \{a\}$.
Then $a \in B$ since $a \in \{s_m\} \cup r_m \cup B$. Then by the definition of E_A, $m_B \in E_A$ so the claim holds.

Case 3: $a \notin \{s_m\} \cup r_m$ and $\neg(A \subseteq \{s_m\} \cup \{a\})$.
If for some note n and groups G and C I have $f(a, n.m, G)_C \in (E_A)_{\leq}$, then either $f(a, n.m, G)_C \in E_A$ or for some $m'_{B'} \in E_A$ I have $f(a, n.m, G)_C < m'_{B'}$. In the first situation I use the fact that the implication $f(a, n.m, G) \to a \blacktriangleleft m$ is valid. In the second situation $m' \to f(a, n.m, G)$ and hence $m' \to a \blacktriangleleft m$. So in both situations the claim holds.

Otherwise let $s'' = s'[m_{B \mapsto B \setminus \{a\}}]$. Note that s'' is a legal state because a does not forward m in s'. Take some $b \in A \setminus (\{s_m\} \cup \{a\})$. Then $s' \sim_b s''$, so $s \sim_A s''$. Moreover, $s'' \models \neg a \blacktriangleleft m$, which yields a contradiction. So this case cannot arise.

($\Leftarrow$) The claim follows directly by the definition of semantics. I provide a proof for one representative case. Suppose that for some email $m'_B \in E_A$ both $A \subseteq S(m') \cup R(m')$ and $m' \rightarrow a \blacktriangleleft m$. Take some legal state s' such that $s \sim_A s'$. Then for some group B' it holds that $m'_{B'} \in E_{s'}$. So $s' \models m'$ and hence $s' \models a \blacktriangleleft m$. Consequently $s \models C_A a \blacktriangleleft m$.

(*ii*) Let $s = (E, N)$.
($\Rightarrow$) Suppose $s \models C_A \neg a \blacktriangleleft m$. Then $s \models \neg a \blacktriangleleft m$. Assume $A \not\subseteq \{s_m\} \cup \{a\}$ and $s \not\models C_A \neg m$. Then there is some legal state $s' = (E', N')$ such that $s \sim_A s'$ and $s' \models m$. Then there is some group B such that $m_B \in E'$. Let $b \in A \setminus (\{s_m\} \cup \{a\})$ and let $s'' = (E' \setminus \{m_B\} \cup \{m_{B \cup \{a\}}\}, N')$. Then $s' \sim_b s''$ so $s \sim_A s''$. But $s'' \models a \blacktriangleleft m$ which contradicts my assumption.

($\Leftarrow$) Suppose that $s \models \neg a \blacktriangleleft m$ and either $A \subseteq \{s_m\} \cup \{a\}$ or $s \models C_A \neg m$. I first consider the case that $A \subseteq \{s_m\} \cup \{a\}$. Let s' be any legal state such that $s \sim_A s'$. Assume $s' \models a \blacktriangleleft m$. Then $m_B \in E_{s'}$ for some group B such that $a \in B$. Since $A \subseteq \{s_m\} \cup \{a\}$, any legal state s'' such that $s' \sim_A s''$ contains an email $m_C \in E_{s''}$ for some group C such that $a \in C$. So $s'' \models a \blacktriangleleft m$. In particular, this holds for the state s, which contradicts my assumption. So $s' \models \neg s(a, n, G)$ and hence $s \models C_A \neg s(a, n, G)$.

Now I consider the case that $s \models C_A \neg m$. Let s' be such that $s \sim_A s'$. Then $s' \models \neg m$. Since $a \blacktriangleleft m \rightarrow m$ is valid, I get $s' \models \neg a \blacktriangleleft m$. So $s \models C_A \neg a \blacktriangleleft m$. $\square$

Now I am ready to prove the Main Theorem.

Proof. (*i*) ($\Rightarrow$) Suppose $s \models C_A m$. Take the legal state s' constructed in Lemma 5.6.1. Then $s \sim_A s'$, so $s' \models m$. So for some group B I have $m_B \in (E_A)_{\leq}$. Hence either $m_B \in E_A$ or some email $m'_{B'} \in E_A$ exists such that $m_B < m'_{B'}$. In both cases the claim holds.

($\Leftarrow$) Suppose that for some email $m'_B \in E_A$ it holds that $m' \rightarrow m$. Take some legal state s' such that $s \sim_A s'$. Then by the form of E_A and the definition of semantics for some group B' I have $m'_{B'} \in E_{s'}$. So $s' \models m'$ and hence $s' \models m$. Consequently $s \models C_A m$.

(*ii*) By the definition of m_B, the fact that the C_A operator distributes over the conjunction, part (*i*) of the Main Theorem and Lemma 5.6.2 I have

$$s \models C_A m_B \text{ iff } \mathbf{C3\text{-}C6},$$

where

C4 $\bigwedge_{a \in \{s_m\} \cup r_m \cup B}$ $((A \subseteq \{s_m\} \cup \{a\}$ and $\exists B' : (m_{B'} \in E_A$ and $a \in B'))$ or $\exists m'_{B'} \in E_A : (m' \rightarrow a \blacktriangleleft m))$,

C5 $\bigwedge_{a \notin \{s_m\} \cup r_m \cup B}$ $(A \subseteq \{s_m\} \cup \{a\}$ or $s \models C_A \neg m)$,

C6 $s \models \bigwedge_{a \notin \{s_m\} \cup r_m \cup B} \neg a \blacktriangleleft m$.

($\Rightarrow$) Suppose $s \models C_A m_B$. Then properties **C3**-**C6** hold. But $|A| \geq 3$ and $s \models C_A m$ imply that no conjunct of **C5** holds. Hence property **C1** holds. Furthermore, since $|A| \geq 3$ the first disjunct of each conjunct in **C4** does not hold. So the second disjunct of each conjunct in **C4** holds, which implies property **C2**.

($\Leftarrow$) Suppose properties **C1**-**C3** hold. It suffices to establish properties **C4**-**C6**. For $a \in \{s_m\} \cup r_m$ I have $m \rightarrow a \blacktriangleleft m$. So **C2** implies property **C4**. Furthermore, since **C1** holds, properties **C5** and **C6** hold vacuously. □

5.7 Analysis of BCC

In this framework I built emails out of messages using the BCC feature. So it is natural to analyze whether and in what sense these emails can be reduced to messages without BCC recipients.

An email with a BCC recipient can be seen as a message without that BCC recipient, followed by a forward by the sender of the message to the BCC recipient. So given a send email $s(a, n, G)_B$, where $B = \{b_1, ..., b_k\}$, it can be simulated with the following sequence of messages:

$$s(a, n, G), f(a, s(a, n, G), b_1), ..., f(a, s(a, n, G), b_k).$$

Analogous simulations can be formed for the forward email $f(a, n.m, G)_B$. At first sight, it seems that this simulation has exactly the same epistemic effect as the original email with the BCC recipients. In both states, each agent $b_1, ..., b_k$ separately receives a copy of the message and only the sender of this message is aware of this. However, there are two subtle differences.

First of all, there is a syntactic difference between the messages that agents $b_1, ..., b_k$ receive in the original case and in the simulation. In the original case they receive exactly the message m, and in the simulation they receive a forward of it. This also means that if they reply to or forward the message, there is a syntactic difference in this reply or forward. This difference is purely syntactic and does not essentially influence the knowledge of the agents, even though it clearly influences the truth value of the formula $b \blacktriangleleft m$ which is true for $b \in \{b_1, ..., b_k\}$ in the original case but not in the simulation.

The second difference is more fundamental. If agents $b_1, ..., b_k$ are BCC recipients of m and they do not send a reply to or a forward of m, then each of them can be sure that no other agent but the sender of m knows he was a BCC recipient. Indeed, in our framework there is no message the sender of m could send to another agent, that expresses that agents $b_1, ..., b_k$ were the BCC recipients of m.

In the case of the simulation however, these recipients do not receive a BCC but a forward. Since these forwards may have additional BCC recipients of which

agents $b_1, ..., b_k$ are unaware, they cannot be sure that the other agents do not know that they received a forward of the message. Furthermore, the sender of m could also forward the forward he sent to $b_1, ..., b_k$ without informing them about it, thus also revealing their knowledge of m.

A concrete example that shows this difference is the following.

5.7.1. Example. Let

$$E_s = \{s(1, n, 2)_{\{3\}}\}.$$

Then $s \models K_3 \neg K_2 K_3 s(1, n, 2)$, that is, agent 3 is sure that agent 2 does not know about his knowledge of the message $s(1, n, 2)$. A simulation of this email without a BCC recipient would result in the state t with (we abbreviate here each email $m_\emptyset$ to m)

$$E_t = \{s(1, n, 2), f(1, s(1, n, 2), 3)\}.$$

Now consider a state t' with:

$$E_{t'} = \{s(1, n, 2), f(1, s(1, n, 2), 3), f(1, f(1, s(1, n, 2), 3), 2)\}.$$

Clearly $t \sim_3 t'$ and $t' \models K_2 K_3 s(1, n, 2)$. This shows that $t \not\models K_3 \neg K_2 K_3 s(1, n, 2)$.

This argument can be made more general as follows. Below, in the context of a state I identify each message m with the email $m_\emptyset$. Then I have the following result.

5.7.2. Theorem. *Take a legal state $s = (E, N)$, an email $m_B \in E$ and an agent $b \in B$ such that E does not contain a forward of m by b or to b. Then for every set of messages (i.e., emails with no BCC recipients) M such that (M, N) is a legal state, I have for every agent $c \notin \{s_m\} \cup \{b\}$*

$$s \models K_b m \wedge K_b \neg K_c K_b m,$$

while

$$(M, N) \not\models K_b m \wedge K_b \neg K_c K_b m.$$

Proof. Agent b is a BCC recipient of m in s, so by the definition of the semantics, $s \models K_b m$. I will first show that $s \models K_b \neg K_c K_b m$. Take some state t such that $s \sim_b t$. Then by the definition of the semantics there is some group C such that $m_C \in E_t$ and $b \in C$. Suppose that m is a send email, say $m = s(a, n, G)$. For the case that m is a forward email the reasoning is analogous. Let u be the state like t, but with

$$E_u = E_t \backslash \{s(a, n, G)_C\} \cup \{s(a, n, G)_{C \backslash \{b\}}, s(a, n, b)\}.$$

Note that I implicitly assume that no full version of $s(a, n, b)$ is already present in E_t. If there were such a full version, I could do the same construction without adding $s(a, n, b)$ to E_t.

Since there are no forwards of m by b or to b in E, and $s \sim_b t$, there are no forwards of m by b or to b in E_t. This shows that u is a legal state and that there are no forwards of m to b in E_u so $u \not\models K_b m$. Clearly, for every $c \notin \{s_m\} \cup \{b\}$ it holds that $t \sim_c u$. So $t \not\models K_c K_b m$, which shows that $s \models K_b \neg K_c K_b m$.

Take now any set of messages M such that (M, N) is legal and suppose $(M, N) \models K_b m$. Then by the Main Theorem there is some message m' in which agent b was involved that implies that message m was sent. By the requirements on the legal states we know that there is such a message m' of which agent b was a recipient, and not the sender, since agents can only send information they initially knew or received through some earlier message. Since there are no BCC recipients in M, I conclude that agent b is a regular recipient of m' and that $m' \to m$ is valid.

Define the set of messages M' by

$$M' := M \cup \{f(s_{m'}, m', c)\}.$$

Note that (M', N) is a legal state, and $(M', N) \models K_c m'$. Since b is a regular recipient of m', $m' \to K_b m'$ is valid and since $m' \to m$ is also valid this implies that $(M', N) \models K_c K_b m$. Also, since b is not involved in $f(S(m'), m', c)$, $(M, N) \sim_b (M', N)$. This shows that $(M, N) \not\models K_b \neg K_c K_b m$. In view of my assumption that $(M, N) \models K_b m$ I conclude that $(M, N) \not\models K_b m \wedge K_b \neg K_c K_b m$. □

In this theorem I assume that for the BCC recipient b of the message m there are no forwards of m to b or by b. The theorem shows that under these assumptions, s and (M, N) can be distinguished by an epistemic formula concerning the message m. I will now show that these assumptions are necessary.

5.7.3. Example. Take a legal state $s = (E, N)$ with

$$E = \{s(1, n, 2)_{\{3\}}, f(2, s(1, n, 2), 3)\}$$

and

$$M = \{s(1, n, 2), f(1, s(1, n, 2), 3), f(2, s(1, n, 2), 3)\}.$$

It is clear that (M, N) is a perfect BCC-free simulation of s: for every formula φ that holds in s, if I replace the occurrences of $3 \blacktriangleleft s(1, n, 2)$ in φ by $f(1, s(1, n, 2), 3)$ then the result holds in (M, N). The reason that I can find such a set M is that in E there is a forward of $s(1, n, 2)$ to agent 3. This reveals the 'secret' that agent 3 knows about $s(1, n, 2)$ and then the fact that agent 3 was a BCC recipient of $s(1, n, 2)$ is no longer relevant.

5.7.4. Example. A similar example shows the importance of the assumption that there are no forwards by a BCC recipient. Take a legal state $s = (E, N)$ with

$$E = \{s(1, n, 2)_{\{3\}}, f(3, s(1, n, 2), 2)\}$$

and

$$M = \{s(1,n,2), f(1,s(1,n,2),3), f(3,f(1,s(1,n,2),3),2)\}.$$

Again, for every formula φ that holds in s, if I replace the occurrences of $3 \blacktriangleleft s(1,n,2)$ in φ by $f(1,s(1,n,2),3)$ then the result holds in (M,N). Now the reason is that agent 3 informed agent 2 that he was a BCC recipient of $s(1,n,2)$ in s by sending a forward of this message, so again the fact that agent 3 knows $s(1,n,2)$ is not a secret anymore.

It is interesting to note that the impossibility of simulating BCC by means of messages is in fact caused by my choice of uninterpreted notes as the basic content of the messages. If my framework allowed one to send messages containing more complex information, for example a formula of the form $b \blacktriangleleft m$, the sender of m could have informed other agents who were the BCC recipients. Then in Example 5.7.1 I could consider a state s' with

$$E_{s'} = \{s(1,n,2)_{\{3\}}, s(1,3 \blacktriangleleft s(1,n,2),2)\}.$$

By appropriately extending the semantics I would then have $s \sim_3 s'$ and $s' \models K_2K_3s(1,n,2)$, and hence $s \not\models K_3\neg K_2K_3s(1,n,2)$, so the difference between the above two states s and t would then disappear.

Similarly, if I allowed epistemic formulas as contents of the messages, then in the above example agent 1 could use the message $s(1,K_3s(1,n,2),2)$ to inform agent 2 that agent 3 was a BCC recipient of the message $s(1,n,2)$. I leave an analysis of such extensions of my framework and the role of BCC in these extended settings as future work.

Finally, let me mention another feature of the syntax that cannot be faithfully simulated by simpler means —that of appending a note to a forwarded message. Suppose that I allow instead only a 'simple' forward $f(i,m,G)$ and simulate the current forward $f(i,n.m,G)$ by a send and a simple forward, i.e., by the sequence $s(i,n,G), f(i,m,G)$. Then the fact that the note n was 'coupled' with m can in some circumstances provide a piece of additional information that becomes lost during the simulation. Here is a concrete example. I do not use BCC here, so each email $m_{\{\emptyset\}}$ is written as m.

5.7.5. Example. Suppose that $n_1, n_2 \in N_1$ and $n_1, n_2 \notin N_a$ for $a \neq 1$. Let $m := s(1,n,1)$ and

$$E_s := \{m, f(1,n_2.m,2)\}.$$

Then for all a it holds that $s \models K_1(K_am \to K_an_2)$, that is, agent 1 knows that every agent who knows the message m also knows the note n_2. A simulation of these two messages with a simple forward would yield the state t with

$$E_t := \{m, s(1,n_2,2), f(1,m,2)\}.$$

Now consider a state t' with:

$$E_{t'} := \{m, s(1, n_2, 2), f(1, m, 2), f(2, m, 3)\}.$$

Clearly $t \sim_1 t'$ and $t' \models K_3 m \wedge \neg K_3 n_2$. This shows that $t \not\models K_1(K_3 m \rightarrow K_3 n_2)$.

Note that this example exploits the fact that in this framework the agents can forward the notes that are 'buried' within the received emails (thanks to the references to $n \in FI(m)$ in conditions L.2 or L.3 in Subsection 5.2.3), whereas they can only forward the messages they received. That is, they cannot forward messages that are 'buried' within the emails they received. This restriction is realistic in the sense that it holds in most email systems.

5.8 Distributed Systems Perspective

In this section I provide a characterization of the notion of a legal state from the perspective of distributed systems. In this setting emails are sent in a nondeterministic order, each time respecting the restrictions imposed by the legality conditions L.1-L.3 of Subsection 5.2.3.

I first define an operational semantics in the style of [Plotkin, 1983], though with some important differences concerning the notions of a program state and the atomic transitions. Let M be the set of all messages (so *not* emails). By a ***mailbox*** I mean a function $\sigma : Ag \rightarrow \mathcal{P}(M)$; $\sigma(a)$ is then the mailbox of agent a. If for all a it holds that $\sigma_0(a) = \emptyset$, then I call σ_0 the ***empty mailbox***. A ***configuration*** is a construct of the form $< s, \sigma >$, where s is a legal state and σ is a mailbox.

Atomic transitions between configurations are of the form

$$< s, \sigma > \quad \rightarrow \quad < s', \sigma' > .$$

Here $\dot{\cup}$ denotes disjoint union and

- $s := (E \,\dot{\cup}\, \{m_B\}, N)$,
- $s' := (E, N)$,
- for $a \in Ag$

$$\sigma'(a) := \begin{cases} \sigma(a) \cup \{m\} & \text{if } a \in r_m \cup \{s_m\} \cup B \\ \sigma(a) & \text{otherwise} \end{cases}$$

I say that the above transition ***processes*** the email m_B. This takes place subject to the following conditions depending on the form of m, where $N = (N_1, \ldots, N_n)$:

- **send** $m = s(a, n, G)$. Then I stipulate that $n \in N_a$ or for some $m' \in \sigma(a)$ it holds that $n \in FI(m')$. In the second case I say that m ***depends on*** m'.
- **forward** $m = f(a, n.m', G)$. Then I stipulate that $m' \in \sigma(a)$, and $n \in N_a$ or for some $m'' \in \sigma(a)$ it holds that $n \in FI(m'')$. In the case of the first alternative I say that m ***depends on*** m' and in the case of the second alternative that m ***depends on*** m' and m''.

These conditions are essentially equivalent to conditions L.1-L.3, as I will show later.

Given a legal state s, an ***email exchange starting in*** s is a maximal sequence of transitions starting in the configuration $< s, \sigma_0 >$, where σ_0 is the empty mailbox. An email exchange ***properly terminates*** if its last configuration is of the form $< s', \tau >$, where $s' = (\emptyset, N)$. The way the atomic transitions are defined clarifies that the communication is synchronous.

Note that messages are never deleted from the mailboxes. Furthermore, observe that in the above atomic transitions I augment the mailboxes of the recipients of m_B (including the BCC recipients) by m and ***not*** by m_B. So the recipients of m_B only 'see' the message m in their mailboxes. Likewise, I augment the mailbox of the sender by the message m and *not* by m_B. As a result when in an email exchange a sender forwards his own email, the BCC recipients of the original email are not shown in the forwarded email. This is consistent with the discussion of the emails given in Subsection 5.2.2.

Observe that from the form of a message m in the mailbox $\sigma(a)$ I can infer whether agent a received it by means of a BCC. Namely, this is the case if and only if $a \notin r_m \cup \{s_m\}$. (Recall that by assumption the sets of regular recipients and BCC recipients of an email are disjoint.)

The following result then clarifies the concept of a legal state.

5.8.1. Theorem. *The following statements are equivalent:*

(i) s is a legal state,

(ii) an email exchange starting in s properly terminates,

(iii) all email exchanges starting in s properly terminate.

The equivalence between (i) and (ii) states that the property of a legal state amounts to the possibility of processing all the emails in an orderly fashion.

Proof. Suppose $s = (E, N)$.

$(i) \Rightarrow (ii)$. Suppose that s is a legal state. So conditions L.1-L.3 are satisfied w.r.t. an spo $\prec$. Extend $\prec$ to a linear ordering $\prec_l$ on E. (Such an extension exists on the account of the result of [Szpilrajn, 1930].) By the definition of the atomic transitions I can process the emails in E in the order determined by $\prec_l$.

The resulting sequence of transitions forms a properly terminating email exchange starting in s.

$(ii) \Rightarrow (iii)$. Let ξ be a properly terminating email exchange starting in s and ξ' another email exchange starting in s. Let m_B be the first email processed in ξ that is not processed in ξ'. The final mailbox of ξ' contains the message(s) that m depends on, since their full versions were processed in ξ before m_B and hence were also processed in ξ'. So m_B can be processed in the final mailbox of ξ', i.e., ξ' is not a maximal sequence. This is a contradiction.

$(iii) \Rightarrow (ii)$. Obvious.

$(ii) \Rightarrow (i)$. Take a properly terminating email exchange ξ starting in s. Take the following spo $\prec$ on the emails of E: $e_1 \prec e_2$ iff e_1 is processed in ξ before e_2. By the definition of the atomic transitions, conditions L.1-L.3 are satisfied w.r.t. $\prec$, so s is legal. □

Intuitively, the equivalence between the first two conditions means that the legality of a state is equivalent to the condition that it is possible to execute its emails in a 'coherent' way. Each terminating exchange entails a strict partial (in fact linear) ordering w.r.t. which conditions L.1-L.3 are satisfied.

5.9 Conclusion

Email is by now one of the most common forms of group communication. This motivates the study presented in this chapter. The language I introduced allowed me to discuss various fine points of email communication, notably forwarding and the use of BCC. The epistemic semantics I proposed aimed at clarifying the knowledge-theoretic consequences of this form of communication. My presentation focused on the issues of epistemic content of the emails and common knowledge.

Communication by email suggests other forms of knowledge. In Chapter 6 I will consider ***potential knowledge*** and ***definitive knowledge*** in the context of email exchanges. When a message is sent to an agent, that agent acquires potential knowledge of it. Only when he forwards the message he acquires definitive knowledge of the message. The idea is that when a message is sent to an agent one cannot be sure that he read it. Only when he forwards it one can be certain that he did read it. The considered framework is an adaptation of the one presented in this chapter. There, common knowledge is not considered but a decision procedure is presented for all considered epistemic formulas.

Another extension worthwhile studying is a setting in which the agents communicate richer basic statements than just notes. I already indicated in Section 6.4 that sending messages containing a formula $a \blacktriangleleft m$ increases the expressiveness of the messages from the epistemic point of view. One could also consider in our framework sending epistemic formulas. One step in this direction is already

present in the approach presented in Chapter 3, where the agents can send each other basic formulas that do not contain epistemic operators. However, there the possible messages are limited to those in a finite set which makes the framework less fit for modeling email communication.

Finally, even though this study was limited to the epistemic aspects of email exchanges, it is natural to suggest here some desired features of emails. One is the possibility of forwarding a message in a provably intact form. This form of forward, used here, is present in the VM email system integrated into the `emacs` editor; in VM forward results in passing the message as an attachment that cannot be changed. Another, more pragmatic one and not considered here, is disabling the reply-all feature for the BCC recipients so that none of them can by mistake reveal that he was a BCC recipient. Yet another one is a feature that would simulate signing of a reception of a registered letter - opening such a 'registered email' would automatically trigger an acknowledgement. Such an acknowledgement would allow one to achieve in a simple way the above mentioned definitive knowledge.

Chapter 6

Possible and Definitive Knowledge in Email Communication

6.1 Introduction

In Chapter 5, I presented a model of the knowledge of agents during an email exchange. Here, I will study the same situation under different assumptions. Instead of focussing on common knowledge, I will distinguish between two different kinds of knowledge: potential knowledge and definitive knowledge.

When an agent receives some information via email, it is possible that he read the email and knows its content. However, one cannot be entirely sure of this because he might have overlooked the email, or he may not have received it at all due to some error in the email system. Therefore, I consider the second agent's knowledge of the email to be potential knowledge. On the other hand, if the agent replies to an email or he forwards it, then he must have read it. In this case I consider the second agent's knowledge to be definitive knowledge. This is relevant in for example a court case, where someone's knowledge of an email may be uncertain if it is only known that someone sent it to him, but his knowledge of the email would be absolutely certain if he also replied to it.

The language presented here is related to the logic presented in Chapter 5. There, the language contains propositions about whether an agent was a BCC recipient of an email and common knowledge modalities, which are not present in the language presented here. Another difference between the languages is that in Chapter 5 there is only one type of knowledge while here I distinguish between potential knowledge and definitive knowledge. Also, in Chapter 5 the only email conversations that are considered are those that are actually possible in the sense that no agent sends information he did not receive. In order to enforce this, certain constraints need to be checked on each email conversation before the analysis takes place. Here, I take a much simpler approach. I do not check whether the email conversation is possible in this sense but just analyze whatever information I can get from it. The advantage of this is that it allows me

to check email conversations of which some emails are not available for analysis.

Another important advantage of the current approach is that I give a finite decision procedure. In Chapter 5 the semantics is only defined by epistemic relations on an infinite number of states. It is unclear whether the model checking of that semantics is possible in finite time, and if it is, the procedure is in any case a lot more complex.

For an overview of existing publications related to this chapter, I refer to Section 5.1.2.

6.1.1 Overview

In the next section, I start out with defining the language based on simple messages with a sender and a set of recipients. I also define a semantics that is given by epistemic relations between sets of these messages. In section 6.3 I show that this semantics can be decided without considering all (possibly infinitely many) epistemically related states. Actual emails also have a list of BCC recipients that is only known to the sender and not to the other recipients. In section 6.4 I add this feature to the semantics and show how it fits in the approach of this chapter.

6.2 The Logic of Messages

In this section I will give a language and semantics based on generic messages with a sender and a set of recipients. In the next section I will focus specifically on emails that also have BCC recipients. I do not analyze the content of messages, only their structure in terms of sender, recipients, and whether they are a forward of or a reply to previous messages. Just like in the previous chapter, I will consider the content of a basic message to be some atomic piece of information that I call a note, usually denoted with n.

Let Ag be a set of agents. I consider messages to have one of two forms:

- A basic message containing a note n, represented by a tuple (a, n, G), where $a \in Ag$ is the sender of the message and $G \subseteq Ag$ is the group of recipients,
- a forward message containing another message, represented by a tuple $(a, n.m, G)$ where $a \in Ag$ is the sender of the message, $G \subseteq Ag$ is the group of recipients, m is some other message and n is a basic note appended to the forward.

I will sometimes leave out the braces from singleton sets, writing for example $(1, n, 2)$ instead of $(1, n, \{2\})$. Given some message m, s_m denotes its sender and r_m the set of its recipients. This set of recipients can be used to model both regular and CC recipients of an email. Note that a reply to a message m can be modeled as (i, m, G) where $s_m \in G$. A reply to all recipients can be modeled as

$(a, m, G \setminus \{a\})$ where $a \in r_m$ and $G = \{s_m\} \cup r_m$. For now, I will assume that the set of recipients is known to the sender and all recipients. In the next section I will also model the BCC recipients of an email.

6.2.1. Example. The expression $(1, n, \{2, 3\})$ stands for a message containing note n from agent 1 to agent 2 and 3. The message $(2, (1, n, \{2, 3\}), \{1, 3\})$ is a reply from agent 2 sent to 1 and 3.

When an agent sends an email to a second agent, the email is usually not read immediately. Sometimes the email is not read at all, for example when it ends up in the spam folder or when the second agent is not very diligent in reading all his emails. Therefore, the first agent cannot be sure that the second agent knows the contents of the email. On the other hand, if the first agent received a reply from the second agent then he is sure the second agent read the email. In the first case, I will say the second agent has ***potential*** knowledge of the email: he may have read it, but then again he may not. In the second case I will say the second agent has ***definitive*** knowledge of the email: since he replied on it, he must have read the email. These two kinds of knowledge are reflected in the following definition.

6.2.2. Definition. The logic of messages and potential and definitive knowledge $\mathcal{L}_{PD}$ is defined as follows:

$$\varphi ::= m \mid \neg\varphi \mid \varphi \wedge \varphi \mid \hat{K}_a\varphi \mid \bar{K}_a\varphi$$

Here m is some message of the form (b, n, G) or (b, m', G) and $a \in Ag$ is some agent.

The formula m expresses the fact that message m was sent. $\hat{K}_a\varphi$ stands for potential knowledge of agent a, which is achieved when agent a receives a message that implies φ. $\bar{K}_a\varphi$ stands for definitive knowledge of agent a, which is achieved when agent a replies to or forwards a message that implies φ. I will use the usual abbreviations $\varphi \vee \psi$ and $\varphi \rightarrow \psi$. Note that knowledge operators may be nested, for example $\bar{K}_a\hat{K}_b m$ expresses that agent a definitively knows that agent b possibly knows m. This may be the case if agent a and b are both recipients of m and agent a forwarded m.

6.2.3. Example. The formula $\hat{K}_2(1, n, \{2, 3\})$ denotes that agent 2 possibly knows that the message $(1, n, \{2, 3\})$ was sent. This is the case whenever this message was sent, because agent 2 is a recipient of it. The formula $\bar{K}_2(1, n, \{2, 3\})$ denotes that agent 2 definitely knows about the message, which is the case when he replied to it.

This language is interpreted on a set of messages M, which I will sometimes call a state. I do not bother to define an ordering between the messages in M. Unlike in the approach presented in Chapter 5, here I do not check whether the set of messages is 'correct' in the sense that for instance no agent forwards a message he did not receive. I just take whatever information is in M and see what I can infer from that. This has the advantage that if not all messages are available for analysis, I can still get the most out of the messages that are available.

In order to really get all information from the messages that are available, even if they are forwards of messages that are themselves not in the set M, I define a closure operation:

6.2.4. Definition. Given a message m or a set of messages M, I define its closure as follows:

$$\begin{array}{lcl} Cl(m) & := & \{m' \mid m' \text{ is mentioned in } m\}, \\ Cl(M) & := & \bigcup_{m' \in M} Cl(m'). \end{array}$$

Just like in the previous chapter, when I say that a message m' is mentioned in another message m I mean that m is m' itself, or a forward of m', or a forward of a forward of m', and so on.

6.2.5. Example. If $M = \{(2, (1, n, \{2, 3\}), \{1, 3\})\}$, then

$$Cl(M) = \{(1, n, \{2, 3\}), (2, (1, n, \{2, 3\}), \{1, 3\})\}.$$

I will now define the semantics of the language $\mathcal{L}_{PD}$. I start out with the first three clauses.

$$\begin{array}{lll} M \models m & \text{iff} & m \in Cl(M) \\ M \models \neg\varphi & \text{iff} & M \not\models \varphi \\ M \models \varphi \wedge \psi & \text{iff} & M \models \varphi \text{ and } M \models \psi \end{array}$$

So I consider M to be evidence for the fact that some message m was sent if m is in the closure of M, that is, if some message in M mentions m.

For the semantics of potential and definitive knowledge of some agent a I will use the perspective of epistemic logic. For every agent, I will define two relations $\sim_a^P$ and $\sim_a^D$ between states, based on the messages in the states. Then I will say that an agent (potentially or definitively) knows a formula in a certain state if that formula holds in all states related to the original state.

For defining these relations $\sim_a^P$ and $\sim_a^D$ between states, I will not look at all messages in M but only to those that agent a sent or received and those that he sent, respectively.

6.2.6. Definition. For each agent a I define two projections on a set of messages M, one for potential knowledge and one for definitive knowledge:

$$\begin{array}{lcl} \Pi_a(M) & := & \{m \in M \mid a \in \{s_m\} \cup r_m\}, \\ \Delta_a(M) & := & \{m \in M \mid a = s_m\}. \end{array}$$

The messages in $\Pi_a(M)$ are exactly those messages for which the fact that they were sent implies that agent a has potential knowledge of this fact. Similarly, the messages in $\Delta_a(M)$ are those messages for which the fact that they were sent implies that agent a has definitive knowledge of that fact.

6.2.7. Example. Let $M = \{(2, n'.(1, n, \{2, 3\}), \{1, 3\})\}$. Then

$$\begin{array}{lcl}
\Delta_1(M) & = & \emptyset, \\
\Pi_2(M) & = & \{(2, (1, n, \{2, 3\}), \{1, 3\})\}, \\
\Pi_3(M) & = & \{(2, (1, n, \{2, 3\}), \{1, 3\})\}, \\
Cl(M) & = & \{(1, n, \{2, 3\}), (2, (1, n, \{2, 3\}), \{1, 3\})\}, \\
\Delta_1(Cl(M)) & = & \{(1, n, \{2, 3\})\}, \\
\Pi_2(Cl(M)) & = & \{(1, n, \{2, 3\}), (2, (1, n, \{2, 3\}), \{1, 3\})\}, \\
\Pi_3(Cl(M)) & = & \{(1, n, \{2, 3\}), (2, (1, n, \{2, 3\}), \{1, 3\})\}.
\end{array}$$

Note that I should first take the closure of M before taking the projection if I want to consider all messages mentioned in M. For example, if I take the projection Δ_1 of M, I do not get the original message sent by agent 1. Only if I first take the closure $Cl(M)$ and then the projection Δ_1 do I get the complete result $\{(1, n, \{2, 3\})\}$. This is correct: agent 1 has definitive knowledge of the message $(1, n, \{2, 3\})$ because he sent it.

Because one should always take the closure before taking a projection, I will define the following shorthand:

6.2.8. Definition. I define:

$$\begin{array}{lcl}
\Pi^*_a(M) & := & Cl(\Pi_a(Cl(M))), \\
\Delta^*_a(M) & := & Cl(\Delta_a(Cl(M))).
\end{array}$$

Now that I have these projections in place, I can continue with defining the relations $\sim^P_a$ and $\sim^D_a$.

6.2.9. Definition. For any two states M and N, I define

$$\begin{array}{lll}
M \sim^P_a N & \text{iff} & \Pi^*_a(M) = \Pi^*_a(N), \\
M \sim^D_a N & \text{iff} & \Delta^*_a(M) = \Delta^*_a(N).
\end{array}$$

With these relations in place, I define the semantics of the knowledge operators as follows:

$$\begin{array}{lll}
M \models \hat{K}_a\varphi & \text{iff} & N \models \varphi \text{ for all } N \text{ such that} M \sim^P_a N \\
M \models \bar{K}_a\varphi & \text{iff} & N \models \varphi \text{ for all } N \text{ such that} M \sim^D_a N
\end{array}$$

Intuitively, this semantics can be interpreted as follows. $\Pi^*_a(M)$ is the 'view' that agent a has on state M, when considering his potential knowledge, that is,

assuming that he read every message that was sent to him. On the other hand, $\Delta_a^*(M)$ is the view of agent a on state M if one considers his definitive knowledge, so assuming that he read only the messages which he replied to or forwarded. Now two states look the same to agent a if his view on them is identical. Therefore, the agent knows something in a certain state if it holds in all states on which he has the same view as on the current state.

Note that the potential knowledge operator and the definitive knowledge operator are *not* each other's dual. It is not necessarily the case that if $M \models \neg\hat{K}_a\neg\varphi$ then also $M \models \bar{K}_a\varphi$, or vice versa.

6.2.10. Example. Again, let $M = \{(2, (1, n, \{2, 3\}), \{1, 3\})\}$. Then

$$\Delta_2^*(M) = \{(1, n, \{2, 3\}), (2, (1, n, \{2, 3\}), \{1, 3\})\}.$$

Because $(1, n, \{2, 3\}) \in \Delta_2^*(M)$, it holds that $M \models \bar{K}_2(1, n, \{2, 3\})$. So in M agent 2 has definitive knowledge of the message $(1, n, \{2, 3\})$. This is correct because agent 2 sent a forward of this message.

For agent 3 this gives:

$$\Delta_3^*(M) = \emptyset.$$

Since $\Delta_3^*(\emptyset) = \emptyset$, it holds that $\emptyset \sim_3^D M$. Because $\emptyset \not\models (1, n, \{2, 3\})$, $M \not\models \bar{K}_3(1, n, \{2, 3\})$. So agent 3 has no definitive knowledge of the message $(1, n, \{2, 3\})$. This is correct because even though agent 3 should have received the original message and the forward by agent 2, he did not reply to this messages or forward them so it is possible that these messages were lost or he did not read them.

I will not give an axiomatization of these semantics. In fact I believe that a complete axiomatization does not exist in the language that is presented here. A complete axiomatization should express the fact that the knowledge of the agents is limited: an agent does not know about a message m if he did not receive some message that mentions m. There is no way to express "there is no message that mentions m" in the language $\mathcal{L}_{PD}$. If there was only a finite number of possible messages then this might be expressed as the negation of a disjunction of messages mentioning m, but since the number of possible messages is unlimited, this cannot be done. Therefore I am convinced that there is no complete axiomatization of the semantics. However, in the next section I will give a way to do model checking of this semantics.

Even though I will give no complete axiomatization, I can give a number of axioms that are valid on all sets of messages under these semantics. They show that the semantics fit the intuition of email communication and possible and definitive knowledge.

6.2.11. THEOREM. *The following axioms hold on all sets of messages:*

$$(a, n.m, G) \rightarrow m \tag{6.1}$$
$$m \rightarrow \bar{K}_a m \qquad (a = s_m) \tag{6.2}$$
$$m \rightarrow \hat{K}_b m \qquad (b \in \{s_m\} \cup r_m) \tag{6.3}$$
$$\bar{K}_a \varphi \rightarrow \hat{K}_a \varphi \tag{6.4}$$

PROOF. Take some set of messages M.
(6.1): Clearly, if $(a, n.m, G) \in Cl(M)$ then $m \in Cl(M)$.
(6.2): If $m \in Cl(M)$ and $a = s_m$ then $m \in \Delta_a(M)$, so $m \in \Delta_a^*(M)$. Let $N \sim_a^D M$. Then $\Delta_a^*(N) = \Delta_a^*(M)$ so $m \in \Delta_a^*(N)$. Then m is mentioned in some $m' \in \Delta_a(Cl(N)) \subseteq Cl(N)$, so $m \in Cl(N)$.
(6.3): The proof is similar to that for (6.2).
(6.4): I will first show that if $M \sim_a^P N$, then $M \sim_a^D N$. Suppose $M \sim_a^P N$. Then $\Pi_a^*(M) = \Pi_a^*(N)$. Take some $m \in \Delta_a^*(M)$. Then m is mentioned in some $m' \in \Delta_a(Cl(M))$. Then $a = s_{m'}$, so certainly $a \in \{s_{m'}\} \cup r_{m'}$. So $m' \in \Pi_a(Cl(M))$ and $m' \in \Pi_a^*(M)$. But then $m' \in \Pi_a^*(N)$. Then m' is mentioned in some $m'' \in \Pi_a(Cl(N)) \subseteq Cl(N)$. So $m' \in Cl(N)$ and because $a = s_{m'}$, $m' \in \Delta_a(Cl(N))$. So because m' mentions m, $m \in \Delta_a^*(N)$. This shows that $\Delta_a^*(M) \subseteq \Delta_a^*(N)$ and analogously I can prove the converse. So $M \sim_a^D N$.
Now suppose $M \models \bar{K}_a \varphi$ and let $M \sim_a^P N$. Then $M \sim_a^D N$ so $N \models \varphi$. Since N was arbitrary this shows that $M \models \hat{K}_a \varphi$. □

6.3 Model Checking

The semantics given in the previous section are very nice in theory. However, can they also be applied in practice? Can it be decided whether a formula holds given some set of messages? It is not complicated to check formulas without epistemic operators. However, when a formula of the form $\hat{K}_i \psi$ or $\bar{K}_i \psi$ needs to be checked in a state M, all states M' with $\Pi_i^*(M) = \Pi_i^*(M')$ or $\Delta_i^*(M) = \Delta_i^*(M')$ have to be checked, respectively. For all we know, there may be infinitely many of these states. In this section I circumvent this problem and I present a way to check formulas with epistemic operators.

6.3.1. DEFINITION. With a literal I mean a message or its negation. If l is a literal, then its negation $\bar{l}$ is $\neg m$ if $l = m$ and m if $l = \neg m$. I call the disjunction of two literals $l \vee l'$ a tautology iff it is of the form $m \vee \neg m'$ (or, equivalently, $\neg m' \vee m$), where $m \in Cl(m')$. I call the disjunction of n literals $l_1 \vee ... \vee l_n$ a tautology iff there are two literals l_i and l_j occurring in that disjunction such that $l_i \vee l_j$ is a tautology. I call the conjunction of n literals $l_1 \wedge ... \wedge l_n$ a contradiction if there are two literals l_i and l_j occurring in that conjunction such that $\overline{l_i} \vee \overline{l_j}$ is a tautology.

It is not hard to see that if $l_1 \vee ... \vee l_n$ is a tautology then for any M, $M \models l_1 \vee ... \vee l_n$. Similarly, if $l_1 \wedge ... \wedge l_n$ is a contradiction then for any M, $M \not\models l_1 \wedge ... \wedge l_n$.

The general idea of my approach is to define for every formula φ a family $\mathcal{F}(\varphi)$ of sets of literals. Then I claim that for any model M, $M \models \varphi$ iff for every $F \in \mathcal{F}(\varphi)$ there is some $l \in F$ such that $M \models l$. One could say that $\mathcal{F}(\varphi)$ represents a conjunctive normal form of φ, using only literals. Because the truth value of literals is easy to check this makes checking the truth value of φ a lot simpler.

So how can any epistemic formula be equivalent to a conjunction of disjunctions of literals? Intuitively, for example the formula $\hat{K}_a m$ can only be true if there was some message sent or received by agent a mentioning message m. Therefore the disjunction of all such messages is a condition for the satisfaction of $\hat{K}_a m$. But because the message sets can contain forwards of forwards of forwards etcetera up to arbitrary depth, there are infinitely many such messages. Therefore, I only consider messages up to a certain depth.

6.3.2. Definition. The depth $\delta(\varphi)$ of a formula φ is defined as follows.

$$\begin{array}{ll} \delta((a,n,G)) & := 1 \\ \delta((a,m,G)) & := 1 + \delta(m) \\ \delta(\neg\psi) & := \delta(\psi) \\ \delta(\psi_1 \wedge \psi_2) & := \max(\delta(\psi_1), \delta(\psi_2)) \\ \delta(\hat{K}_a\psi) & := 1 + \delta(\psi) \\ \delta(\bar{K}_a\psi) & := 1 + \delta(\psi) \end{array}$$

The depth of a set of messages M is defined as $\delta(M) := \max(\{\delta(m) \mid m \in M\})$. Note that if $m \in Cl(m')$ then $\delta(m) \leq \delta(m')$. This implies that for any M, $\delta(M) = \delta(Cl(M))$.

I will construct $\mathcal{F}(\varphi)$ with literals up to a certain depth. I will later show that for any state and formula a bound can be found on the depth of the literals that need to be considered.

6.3.3. Definition. Given a message m, let $\mathcal{M}^n_{Ag}(m)$ be the set of all possible messages m' of depth $\leq n$ between the agents in Ag such that $m \in Cl(m')$.

6.3.4. Definition. Let φ be a formula with $\delta(\varphi) \leq n$. I define a family of sets of literals $\mathcal{F}^n(\varphi)$ as follows. For $\varphi = m$, let

$$\mathcal{F}^n(m) := \{\{m\}\}.$$

For $\varphi = \neg\psi$, suppose $\mathcal{F}^n(\psi) = \{F_1, ..., F_n\}$. Then

$$\mathcal{F}^n(\neg\psi) := \{\{\overline{l_1}, ..., \overline{l_n}\} \mid l_1 \in F_1, ..., l_n \in F_n\}.$$

For $\varphi = \psi_1 \wedge \psi_2$, let

$$\mathcal{F}^n(\psi_1 \wedge \psi_2) := \mathcal{F}^n(\psi_1) \cup \mathcal{F}^n(\psi_2).$$

For $\varphi = \hat{K}_a\psi$, let

$$\begin{aligned}\mathcal{F}^n(\hat{K}_a\psi) := \{ \quad & \{m \in F \mid a \in \{s_m\} \cup r_m\} \cup \\ & \{m' \in \mathcal{M}^n_{Ag}(m) \mid m \in F, a \notin \{s_m\} \cup r_m, a \in \{s_{m'}\} \cup r_{m'}\} \cup \\ & \{\neg m' \mid \neg m \in F, m' \in Cl(m), a \in \{s_{m'}\} \cup r_{m'}\} \ \mid F \in \mathcal{F}^n(\psi)\}\end{aligned}$$

For $\varphi = \bar{K}_a\psi$, let

$$\begin{aligned}\mathcal{F}^n(\bar{K}_a\psi) := \{ \quad & \{m \in F \mid a = s_m\} \cup \\ & \{m' \in \mathcal{M}^n_{Ag}(m) \mid m \in F, a \neq s_m, a = s_{m'}\} \cup \\ & \{\neg m' \mid \neg m \in F, m' \in Cl(m), a = s_{m'}\} \ \mid F \in \mathcal{F}^n(\psi)\}\end{aligned}$$

I will explain this definition step by step. The definition for $\varphi = m$ is obvious: clearly, $M \models m$ iff there is some $l \in \{m\}$ such that $M \models l$. For $\varphi = \neg\psi$, note that $M \models \neg\psi$ if $M \not\models \psi$, so if there is $F \in \mathcal{F}^n(\psi)$ such that for any $l \in F$, $M \models \bar{l}$. But this is exactly the case if there is for any $F' \in \mathcal{F}^n(\neg\psi)$ some $\bar{l} \in F'$ such that $l \in F$ and $M \models \bar{l}$. For $\varphi = \psi_1 \wedge \psi_2$, note that the necessary condition holds for every $F_1 \in \mathcal{F}^n(\psi_1)$ and for every $F_2 \in \mathcal{F}^n(\psi_2)$ iff it holds for every $F \in \mathcal{F}^n(\psi_1) \cup \mathcal{F}^n(\psi_2)$.

For $\varphi = \hat{K}_a\psi$, I consider every literal in some member of $\mathcal{F}^n(\psi)$ separately. If it is a message m such that $a \in \{s_m\} \cup r_m$ then m is equivalent to $\hat{K}_a m$ so I preserve m in some member of $\mathcal{F}^n(\varphi)$. If it is a message m with $a \notin \{s_m\} \cup r_m$ then agent a has possible knowledge of m if some forward or a forward of a forward etcetera was sent by or to agent a. Therefore I replace m by all members of $\mathcal{M}^n_{Ag}(m)$ which were sent to or by agent a. Note that here I only consider messages of depth $\leq n$. For the case that the literal is the negation of a message $\neg m$, note that agent a knows that m was not sent if there is some message mentioned in m of which he was a sender or a recipient, which was not sent. Therefore I replace $\neg m$ with these messages.

The definition for $\varphi = \bar{K}_a\psi$ is very similar to that for $\hat{K}_a\psi$, only now I only look at messages sent by agent a, instead of those sent or received by agent a.

The following theorem states that for every model and formula, I can find a number such that the satisfaction of that formula in that model can be decided by looking at a family of sets of literals of depth up to that number:

6.3.5. THEOREM. *For any set of messages M and formula φ there is a finite number $n_{M,\varphi} \geq \delta(M)$ such that for every $k \geq n_{M,\varphi}$,*

$$M \models \varphi \textit{ iff any } F \in \mathcal{F}^k\varphi \textit{ contains a literal } l \in F \textit{ such that } M \models l.$$

PROOF. See Section 6.7. □

Now I can check whether a formula φ holds in a state M by only considering the literals in $\mathcal{F}^{n_{M,\varphi}}(\varphi)$. However, I have no idea how large $n_{M,\varphi}$ will be, and I may have to check a very large number of literals. This apparent problem quickly disappears with the following realisation. For any message m with $\delta(m) > \delta(M)$, certainly $M \models \neg m$. So I can remove any m with $\delta(m) > \delta(M)$ from any member of $\mathcal{F}^{n_{M,\varphi}}(\varphi)$. Also, any member of $\mathcal{F}^{n_{M,\varphi}}(\varphi)$ that contains some literal $\neg m$ with $\delta(m) > \delta(M)$ can be removed altogether, because certainly $M \models \neg m$.

6.3.6. DEFINITION. Given a formula φ and two numbers $n > k$, I define the restriction of $\mathcal{F}^n(\varphi)$ to depth k as follows:

$$\mathcal{F}^n(\varphi)|k := \{ \quad \{l \in F \mid \delta(l) \leq k\} \mid F \in \mathcal{F}^n(\varphi), F \text{ contains no } \neg m \text{ such that } \delta(m) > k\}.$$

6.3.7. THEOREM. *For any state M, formula φ and number $n > \delta(M)$, there is for every $F \in \mathcal{F}^n(\varphi)$ some $l \in F$ such that $M \models l$, if and only if the same holds for $\mathcal{F}^n(\varphi)|\delta(M)$.*

PROOF. Suppose for every $F \in \mathcal{F}^n(\varphi)$ there is some $l \in F$ such that $M \models l$. Take some $F' \in \mathcal{F}^n(\varphi)|\delta(M)$ and let F be the set on which F' is based. Take $l \in F$ such that $M \models l$. Because $M \models l$, either $l = \neg m$ for some message m with $\delta(m) > \delta(M)$ or $\delta(l) \leq \delta(M)$. In the first case, $F' \notin \mathcal{F}^n(\varphi)$ by definition so this is not possible. In the second case, $l \in F'$ so the requirement is satisfied for F'.

Conversely, suppose for every $F' \in \mathcal{F}^n(\varphi)|\delta(M)$ there is some $l \in F'$ such that $M \models l$. Take some $F \in \mathcal{F}^n(\varphi)$. Suppose there is $\neg m \in F$ such that $\delta(m) > \delta(M)$. Then $M \models \neg m$ so the requirement is satisfied for F. Suppose there is no such $\neg m \in F$. Then there is $F' \in \mathcal{F}^n(\varphi)|\delta(M)$ based on F. Then there is some $l \in F'$ such that $M \models l$. But $F' \subseteq F$ so then $l \in F$ and the requirement is satisfied for F. □

This theorem already reduces the collection of literals that need to be checked to those of depth $\leq \delta(M)$. Furthermore, checking the truth value of these literals can be optimized in many ways. In many cases a disjunction of all possible messages with a certain sender or recipient will need to be checked, so a data structure that indexes the messages in a state by the agents involved in them might help a lot. All in all, I am convinced that this semantics is a promising basis for an efficient model checker of the language $\mathcal{L}_{PD}$.

6.4 Blind Carbon Copy

In this section I will extend my semantics to an approach specifically tailored to emails. The difference between the earlier messages and emails is that emails

have a set of BCC recipients. These BCC recipients receive the email as well, but this fact is only known to the sender of the email.

Just like in Chapter 5 I define an email to be a construct of the form $e = m_B$, where m is a message as defined in the previous section and $B \subseteq Ag$ is a set of BCC recipients. I will use s_e, r_e and $B(e)$ to denote the sender, the set of regular recipients and the set of BCC recipients of an email e. So if $e = m_B$, then $s_e = s_m$, $r_e = r_m$ and $B(e) = B$. Given an email $e = m_B$ I will say that e is based on the message m. I will identify a message without a set of BCC recipients that is a member of a set of emails $m \in E$ with the same message with an empty set of BCC recipients: $m_\emptyset$.

Just like in reality, the BCC recipients of a message that is forwarded are not mentioned in the forward. So a forward of an email m_B is an email of the form $(i, m, G)_C$. Note that B is not mentioned in the forward.

I do not change the language with the addition of BCC recipients. This means that the BCC recipients are not mentioned in the logic at all. This differs from the approach presented in Chapter 5, where an extra language construct is introduced in order to make the BCC recipients explicit in the language. However, I will show that it is very well possible to analyze the agents' knowledge in a situation with BCC recipients without mentioning them explicitly in the language.

Let E be some set of emails. Just like in the previous section, I will define the closure of the set E. However, this becomes a bit more complicated because I have to take the BCC recipients into account. The following example shows how this complicates matters.

6.4.1. Example. Suppose Alice sends an email to Bob, with a BCC to Carol. Then Bob does not know that Carol received the message. However, now Carol sends a reply to this email to both Alice and Bob. Then Bob gets to know that Carol received the original email. By sending the reply, Carol revealed her identity as a BCC recipient.

Formalizing this example, let agent 1 be Alice, agent 2 be Bob and agent 3 be Carol. The original email would be formalized as $(1, n, 2)_3$ and the reply by Carol as $(3, (1, n, 2), \{1, 2\})$. From the second email it can be deduced that 3 was a BCC recipient of the first email. Therefore, the closure of the set $\{(3, (1, n, 2), \{1, 2\})\}$ should include the message $(1, n, 2)$ *with* a BCC to agent 3, even though this BCC recipient is not mentioned explicitly.

In order to define the closure, I first compute for each message its BCC recipients, according to some set of emails.

$$\begin{aligned} B(m, E) := \{ \quad & b \in Ag \backslash (\{s_m\} \cup r_m) \mid \\ & \exists C : m_C \in E \text{ and } b \in C \text{ or} \\ & \exists G : (b, m, G) \text{ is mentioned in some } e \in E\} \end{aligned}$$

So an agent b is in $B(m, E)$ if it can be deduced from the set E that b was a BCC recipient of E. This is the case if there is some email m_C in E that shows that b

was a BCC recipient because $b \in C$, or if b forwarded m to some other group of agents.

Using this definition I define the closure of a set of emails as follows:

$$Cl(E) := \{m_{B(m,E)} \mid \exists e \in E : m \text{ is mentioned in } e\}$$

So I take any message that is mentioned in some email in E, and add the BCC recipients that can be deduced from E.

Now that I have defined the closure of a set of emails, I should also define the projections for the agent's knowledge. In order to simplify the definitions, I first define a new notion of union that takes BCC recipients into account:

$$\begin{aligned} E \cup^* E' \;:=\; & \{m_B \in E \mid \neg\exists B' : m_{B'} \in E'\} \cup \\ & \{m_{B'} \in E' \mid \neg\exists B : m_B \in E\} \cup \\ & \{m_{B \cup B'} \mid m_B \in E, m_{B'} \in E'\} \end{aligned}$$

This notion of union is designed to make sure that if a message occurs in both E and E' with different BCC recipients, the BCC recipients are joined in one set instead of including the message twice.

I continue with the projection for potential knowledge. In this definition I carefully make out which BCC recipients of each email are visible to the agent. If the agent is the sender of the email, all BCC recipients are visible to him. If he is a regular recipient and not a BCC recipient, then none are visible. If he is a BCC recipient himself, then he only knows that he himself is a BCC recipient and he does not know the identity of any other BCC recipients.

$$\begin{aligned} \Pi_a(E) := \; & \{m_B \in E \mid a = s_m\} \cup^* \\ & \{m_\emptyset \mid \exists B : m_B \in E, a \in r_m\} \cup^* \\ & \{m_{\{a\}} \mid \exists B : m_B \in E, a \in B\} \end{aligned}$$

Note that in this definition I ignore what the agent can deduce about the BCC recipients of an email by looking at forwards sent by those BCC recipients. That is why, after applying a projection, I will always take the closure of the result.

Now I turn to the projection for definitive knowledge. This is quite simple: since I only look at emails where the agent is the sender, all BCC recipients are visible to him so they are all preserved by the projection.

$$\Delta_a(E) := \{m_B \in E \mid a = s_m\}$$

Again, I define a shorthand for taking the projection and the closure:

$$\Pi_a^*(E) := Cl(\Pi_a(Cl(E))),$$

$$\Delta_a^*(E) := Cl(\Delta_a(Cl(E))).$$

Note that if one views a message as an email with an empty set of BCC recipients, then the new definitions for closure and projections coincide with the ones given in Section 6.2.

The semantics of the language on sets of emails is defined in the same way as for sets of messages. I define that $E \sim_i^P E'$ iff $\Pi_i^*(E) = \Pi_i^*(E')$, and similarly for $\sim_i^D$ and Δ_i^*. Then the semantics for sets of emails is given by:

$$\begin{array}{lll} E \models m & \text{iff} & \exists B : m_B \in Cl(E) \\ E \models \neg\varphi & \text{iff} & E \not\models \varphi \\ E \models \varphi \wedge \psi & \text{iff} & E \models \varphi \text{ and } E \models \psi \\ E \models \hat{K}_a\varphi & \text{iff} & E' \models \varphi \text{ for all } E' \text{ such that } E \sim_a^P E' \\ E \models \bar{K}_a\varphi & \text{iff} & E' \models \varphi \text{ for all } E' \text{ such that } E \sim_a^D E' \end{array}$$

The following example shows how this semantics works out.

6.4.2. Example. Suppose agent 1 sends an email to agent 2, with a BCC to 3 and 4. Then agent 3 forwards this email to agent 2. I formalize this as follows:

$$\begin{array}{rcl} E & = & \{(1,n,2)_{\{3,4\}}, (3,(1,n,2),2)\}, \\ Cl(E) & = & E. \end{array}$$

In order to analyze the knowledge of agent 3, I compute the projections $\Pi_3^*(E)$ and $\Delta_3^*(E)$:

$$\begin{array}{rcl} \Pi_3^*(E) & = & \{(1,n,2)_3, (3,(1,n,2),2)\}, \\ \Delta_3^*(E) & = & \{(1,n,2)_3, (3,(1,n,2),2)\}. \end{array}$$

Because $(1,n,2)_3 \in \Pi_3^*(E)$, it holds that $E \models \hat{K}_3(1,n,2)$. This was to be expected: agent 3 possibly knows about the email $(1,n,2)$ because he received a BCC of it.
Because $(3,(1,n,2),2) \in \Delta_3(Cl(E))$ it holds that $(1,n,2)_3 \in \Delta_3^*(E)$ and $E \models \bar{K}_E(1,n,2)$. Intuitively speaking, agent 3 definitively knows about $(1,n,2)$ because he sent a forward of it.
Now I consider the knowledge of agent 4 about agent 3's knowledge:

$$\begin{array}{rcl} \Pi_4^*(E) & = & \{(1,n,2)_4\}, \\ \Pi_3^*(\Pi_4^*(E)) & = & \emptyset. \end{array}$$

Because $(1,n,2) \notin \Pi_3^*(\Pi_4^*(E))$, it holds that $E \models \neg\hat{K}_4\hat{K}_3(1,n,2)$. So agent 4 does not know that 3 knows about the first email. This is because agent 4 does not know that 3 was also a BCC recipient. However, agent 1 does know this, as is shown by the following projections:

$$\begin{array}{rcl} \Pi_1^*(E) & = & \{(1,n,2)_{\{3,4\}}\}, \\ \Pi_3^*(\Pi_1^*(E)) & = & \{(1,n,2)_3\}, \\ \Delta_3^*(\Pi_1^*(E)) & = & \emptyset. \end{array}$$

Because agent 1 is the sender of the first email, agent 3 is preserved as a BCC recipient in the projection $\Pi_1^*(E)$. Then when I take the potential knowledge projection for agent 3 the original message is again preserved so $(1, n, 2)_{\{3\}} \in \Pi_3^*(\Pi_1^*(E))$. Therefore, $E \models \hat{K}_1\hat{K}_3(1, n, 2)$.
However, the forward by agent 3 is not in $\Pi_1^*(E)$, nor is any other email sent by agent 3, so when I take the definitive knowledge projection for agent 3 then the result is the empty set: $\Delta_3^*(\Pi_1^*(E)) = \emptyset$. Therefore, $(1, n, 2) \notin \Delta_3^*(\Pi_1^*(E))$ and $E \models \neg\hat{K}_1\bar{K}_3(1, n, 2)$: agent 1 does not know that agent 3 definitively knows about the original message, because he did not receive agent 3's forward. This means that agent 1 cannot be entirely sure that his email actually reached agent 3. Agent 2, on the other hand, did receive agent 3's forward. Let me consider the projections for agent 2:

$$\begin{aligned}
\Pi_2(Cl(E)) &= \{(1, n, 2), (3, (1, n, 2), 2)\},\\
\Pi_2^*(E) &= \{(1, n, 2)_3, (3, (1, n, 2), 2)\},\\
\Pi_3(Cl(\Pi_2(Cl(E)))) &= \{(1, n, 2)_3, (3, (1, n, 2), 2)\},\\
\Pi_3^*(\Pi_2^*(E)) &= \{(1, n, 2)_3, (3, (1, n, 2), 2)\},\\
\Delta_3^*(\Pi_2^*(E)) &= \{(1, n, 2)_3, (3, (1, n, 2), 2)\}.
\end{aligned}$$

When I take the projection $\Pi_2(Cl(E))$, then initially no BCC recipients of $(1, n, 2)$ are preserved because as a regular recipient, agent 2 does not know the identity of the BCC recipients. However, because agent 3 forwarded the email $(1, n, 2)$, agent 2 knows that agent 3 was a BCC recipient. This is reflected by the fact that in the closure $Cl(\Pi_2(Cl(E)))$, agent 3 is a BCC recipient of $(1, n, 2)$. This shows exactly why it is important to apply the closure after applying a projection.

Because 3 is a BCC recipient of $(1, n, 2)$ in $\Pi_2^*(E)$, the message $(1, n, 2)$ is preserved in $\Pi_3^*(\Pi_2^*(E))$, and because of that

$$E \models \hat{K}_2\hat{K}_3(1, n, 2).$$

Something even stronger can be said: because

$$(3, (1, n, 2), 2) \in \Pi_2^*(E)$$

it also holds that

$$(1, n, 2)_3 \in \Delta_3^*(\Pi_2^*(E)),$$

which means that

$$E \models \hat{K}_2\bar{K}_3(1, n, 2).$$

Intuitively, agent 2 knows that agent 3 definitely knows about the first message because he received the forward by agent 3.

6.5 Model Checking with BCC

Now that I have extended the semantics with BCC, I can ask again the question of whether it is possible to do model checking of the semantics in finite time. I think this is certainly possible.

When a message m has to be sent with a set of BCC recipients B, this can be done as an email m_B. But another option is for the sender of m to first send the message m, and then send a forward (s_m, m, b) for every $b \in B$. This is the simulation I already mentioned in Chapter 5. I will make this formal as follows.

6.5.1. Definition. Given a message m, let $\beta(m)$ be the message constructed from m by replacing all occurrences in m of some message (b, m', G) where $b \notin \{s_{m'}\} \cup r_{m'}$ by the message $(b, (s_{m'}, m', b), G)$. Similarly, for a formula φ, $\beta(\varphi)$ is constructed by replacing all occurrences of messages m in φ by $\beta(m)$.

So if some agent forwarded a message of which he was not the sender or a regular recipient, in which case he must have been a BCC recipient, then I replace the forward by a forward of a forward by the sender of the first message. Using this transformation β I can transform a set of emails to a set of messages as follows:

6.5.2. Definition. Given a set of emails E, I construct $\beta(E)$ by replacing each email m_B with the messages in

$$\{m\} \cup \{(s_m, m, b) \mid b \in B\}$$

and subsequently replacing every message m in the result by $\beta(m)$.

This transformation can be interchanged with the application of the projection.

6.5.3. Lemma (22). *For any set of emails E and any agent a,*

$$\beta(\Pi_a(E)) = \Pi_a(\beta(E)).$$

Similarly for Δ_a.

Proof. Take some $m \in \beta(\Pi_a(E))$.

Suppose $m = \beta(m^1)$ for some $m^1_B \in \Pi_a(E)$. Then there is $m^2_C \in \Pi_a(Cl(E))$ mentioning m^1. Then $a \in \{s_{m^2}\} \cup r_{m^2} \cup C$ and m^2 is mentioned in some $m^3_D \in E$. Then $\beta(m^3) \in \beta(E)$ and $\beta(m^2)$ is mentioned in $\beta(m^3)$ so $\beta(m^2) \in Cl(\beta(E))$. Suppose $a \in s_{m^2} \cup r_{m^2}$. Then $\beta(m^2) \in \Pi_a(Cl(\beta(E)))$ and because $\beta(m^1)$ is mentioned in $\beta(m^2)$ then $m \in \Pi^*_a(\beta(E))$. Suppose $a \in C$. Then $a \in B(m^2, E)$. Then either there is some set C' such that $a \in C'$ and $m^2_{C'} \in E$ or there is some group G such that $(a, m^2, G) \in E$. Suppose the first case. Then $(s_{\beta(m^2)}, \beta(m^2), a) \in \beta(E)$

so $(s_{\beta(m^2)}, \beta(m^2), a) \in \Pi_a(Cl(\beta(E)))$ and $\beta(m^1) \in \Pi_a^*(\beta(E))$. Suppose the second case. Then $(a, (s_{m^2}, \beta(m^2), a), G) \in \beta(E)$ so $(a, (s_{m^2}, \beta(m^2), a), G) \in \Pi_a(Cl(\beta(E)))$ and $\beta(m^1) \in \Pi_a^*(\beta(E))$.

Suppose $m = (s_{m'}, \beta(m'), b)$ for some $m'_B \in \Pi_a(E)$ with $b \in B$. Then $b \in B(m', \Pi_a(Cl(E)))$. Suppose there is C such that $b \in C$ and $m'_C \in \Pi_a(Cl(E))$. Then $a \in \{s_{m'}\} \cup \{b\}$ and $b \in B(m', E)$. Suppose there is D with $b \in D$ and $m'_D \in E$. Then $(s_{m'}, \beta(m'), b) \in \beta(E)$. Since $a \in \{s_{m'}\} \cup \{b\}$ then $(s_{m'}, \beta(m'), b) \in \Pi_a^*(\beta(E))$. Suppose there is no such D. Then (b, m', G) is mentioned in $Cl(E)$ for some group G. Then $(b, (s_{m'}, \beta(m'), b), G) \in Cl(\beta(E))$ and because $a \in \{s_{m'}\} \cup \{b\}$ then $(b, (s_{m'}, \beta(m'), b), G) \in \Pi_a(Cl(\beta(E)))$ so $(s_{m'}, \beta(m'), b) \in \Pi_a^*(\beta(E))$. Now suppose there is no such C. Then there is G' such that (b, m', G') is mentioned in $\Pi_a(Cl(E))$. By a similar reasoning as above then $(b, (s_{m'}, \beta(m'), b), G') \in \Pi_a(Cl(\beta(E)))$ so $m \in \Pi_a^*(\beta(E))$.

For the converse, take some $m \in \Pi_a(\beta(E))$. Then there is some $m' \in \Pi_a(Cl(\beta(E)))$ mentioning m. Then $a \in \{s_{m'}\} \cup r_{m'}$ and m' is mentioned in some $m'' \in \beta(E)$. Suppose $m'' = \beta(m^1)$ for some $m^1_B \in E$. Then there is some m^2 mentioned in m^1 such that $m' = \beta(m^2)$. Then $a \in \{s_{m^2}\} \cup r_{m^2}$ so $m^2_C \in \Pi_a(Cl(E))$ for some C. Then there is some m^3 mentioned in m^2 such that $m = \beta(m^3)$. So then there is some D such that $m^3_D \in \Pi_a^*(E)$ and $m \in \beta(\Pi_a^*(E))$.

Now suppose $m'' = (s_{m^1}, \beta(m^1), b)$ for some $m^1_B \in E$ with $b \in B$. Then there is m^2 mentioned in m^1 such that $m' = \beta(m^2)$. Because $a \in \{s_{m'}\} \cup r_{m'}$ then there is some C such that $m^2_C \in \Pi_a(Cl(E))$. Then there is some m^3 mentioned in m^2 such that $m = \beta(m^3)$. Then $m_3 \in \Pi_a^*(E)$ so $m \in \beta(\Pi_a^*(E))$. □

In Chapter 5, two differences between the original email m_B and the simulation with forwards are mentioned. The first one is that every agent in B receives a forward of m instead of m itself. This syntactic difference is preserved when the agents in B forward the message or the forward of the message. However, it does not influence the agent's knowledge about m or about each other's knowledge of m.

The second difference is that when an agent is a BCC recipient, and he does not reveal this fact to others by sending a forward, then he knows that the other agents do *not* know he received the message. This is because the BCC recipients are not included in forwards of the original message. On the other hand, if the sender of the message sent a separate forward to the former BCC recipient then the sender may forward this forward to other agents, thereby informing them that the former BCC recipient knows about the message. In other words, the BCC feature makes the fact that these agents receive the message a secret, while a separate forward does not.

This may seem contradictory to Lemma 6.5.3 because it seems that that result implies that the transformation β does not influence the knowledge relations. This apparent contradiction is caused by the fact that it is possible that there are two sets of emails E and E' such that $\Pi_a(\beta(E)) = \Pi_a(\beta(E'))$ while $\Pi_a(E) \neq \Pi_a(E')$.

Then, clearly $\beta(E) \sim_a \beta(E')$ while $E \not\sim_a E'$. The following example shows how this can occur.

6.5.4. Example. Consider the following sets of emails:

$$\begin{array}{ll} E_1: & \{(1,n,2)_3\} \\ E_2: & \{(1,n,2),(1,(1,n,2),3),(1,(1,(1,n,2),3),2)\} \end{array}$$

Then $E_1 \not\sim_3 E_2$. Note that $E_2 \models \hat{K}_2\hat{K}_3(1,n,2)$ while $E_1 \not\models \hat{K}_2\hat{K}_3(1,n,2)$. In fact, there is no E' such that $E_1 \sim_3 E'$ and $E' \models \hat{K}_2\hat{K}_3(1,n,2)$, so $E_1 \models \hat{K}_3\neg\hat{K}_2\hat{K}_3(1,n,2)$.

Now look at the transformed sets of emails:

$$\begin{array}{lcl} \beta(E_1) & = & \{(1,n,2),(1,(1,n,2),3)\} \\ \beta(E_2) & = & \{(1,n,2),(1,(1,n,2),3),(1,(1,(1,n,2),3),2)\} \end{array}$$

I have $\beta(E_1) \sim_3 \beta(E_2)$. However, $\beta(E_2) \models \hat{K}_2\hat{K}_3(1,n,2)$ so

$$\beta(E_3) \not\models \hat{K}_3\neg\hat{K}_2\hat{K}_3(1,n,2).$$

This shows that even though the β transformation gives a good simulation of a set of emails without using BCC, it is not perfect. In other words, BCC really adds something new from an epistemic perspective. Therefore, for deciding the model checking problem with BCC it is not enough to simply translate the sets of emails to sets of messages and handle the model checking as in Section 6.3.

A better way to solve the model checking problem would be to adapt the definition of $\mathcal{F}^n(\varphi)$ from the previous section for the case with BCC recipients. This new definition of $\mathcal{F}^n(\varphi)$ will have the same function as for the semantics without BCC. However, now the sets in $\mathcal{F}^n(\varphi)$ will not only contain literals, but also constructs of the form m_j and negations of these constructs. Here m is a message and j is a single agent. The satisfaction of these constructs in a state is defined as follows:

$$E \models m_b \text{ iff there is } B \subseteq Ag : m_B \in E, b \in B.$$

Note that I do not want to extend the logic with this new construct m_j. I only use it to decide the truth value of the formulas.

I continue with the new definition of $\mathcal{F}^n(\varphi)$.

6.5.5. Definition. Let φ be a formula with $\delta(\varphi) \leq n$. I define a family of sets of literals $\mathcal{F}^n(\varphi)$ as follows. For $\varphi = m$, let

$$\mathcal{F}^n(m) := \{\{m\}\}$$

For $\varphi = \neg\psi$, suppose $\mathcal{F}^n(\psi) = \{F_1, ..., F_n\}$. Then

$$\mathcal{F}^n(\neg\psi) := \{\{\overline{l_1}, ..., \overline{l_n}\} \mid l_1 \in F_1, ..., l_n \in F_n\},$$

where $\bar{l}$ is given by $\neg m$ if $l = m$ and m if $l = \neg m$. For $\varphi = \psi_1 \wedge \psi_2$, let

$$\mathcal{F}^n(\psi_1 \wedge \psi_2) := \mathcal{F}^n(\psi_1) \cup \mathcal{F}^n(\psi_2).$$

For $\varphi = \hat{K}_a\psi$, let

$$\mathcal{F}^n(\hat{K}_a\psi) := \{\bigcup_{l \in F} \mathbf{F}^n_{\hat{K}_a}(l) \mid F \in \mathcal{F}^n(\psi)\},$$

where $\mathbf{F}^n_{\hat{K}_a}(l)$ is given by

$$\begin{array}{ll}
\{m\} & \text{if} \quad l = m, a \in \{s_m\} \cup r_m, \\
\{m' \in \mathcal{M}^n_{Ag}(m) \mid a \in \{s_{m'}\} \cup r_{m'}\} \cup & \\
\{m'_a \mid m' \in \mathcal{M}^n_{Ag}(m)\} & \text{if} \quad l = m, a \notin \{s_m\} \cup r_m, \\
\{\neg m' \mid m' \in Cl(m), a \in \{s_{m'}\} \cup r_{m'}\} & \text{if} \quad l = \neg m, \\
\{m_b\} & \text{if} \quad l = m_b, a \in \{s_m\} \cup \{b\}, \\
\{\neg m_b\} & \text{if} \quad l = \neg m_b, a \in \{s_m\} \cup \{b\}, \\
\{(b, m, G) \mid G \subseteq Ag, a \in G\} \cup & \\
\{m' \in \mathcal{M}^n_{Ag}((b, m, G)) & \\
\quad \mid G \subseteq Ag, a \in \{s_{m'}\} \cup r_{m'}, a \notin G\} \cup & \\
\{m'_a \mid m' \in \mathcal{M}^n_{Ag}((b, m, G)), G \subseteq Ag, a \notin G\} & \text{if} \quad l = m_b, a \notin \{s_m\} \cup \{b\}, \\
\{\neg m' \mid m' \in Cl(m), a \in \{s_{m'}\} \cup r_{m'}\} & \text{if} \quad l = \neg m_b, a \notin \{s_m\} \cup \{b\}.
\end{array}$$

For $\varphi = \bar{K}_a\psi$, let

$$\mathcal{F}^n(\bar{K}_a\psi) := \{\bigcup_{l \in F} \mathbf{F}^n_{\bar{K}_a}(l) \mid F \in \mathcal{F}^n(\psi)\},$$

where $\mathbf{F}^n_{\bar{K}_a}(l)$ is given by

$$\begin{array}{ll}
\{m\} & \text{if} \quad l = m, a = s_m, \\
\{m' \in \mathcal{M}^n_{Ag}(m) \mid a = s_{m'}\} & \text{if} \quad l = m, a \neq s_m, \\
\{\neg m' \mid m' \in Cl(m), a = s_{m'}\} & \text{if} \quad l = \neg m, \\
\{m_b\} & \text{if} \quad l = m_b, a = s_m, \\
\{m' \in \mathcal{M}^n_{Ag}((b, m, G)) \mid a = s_{m'}\} & \text{if} \quad l = m_b, a \neq s_m, \\
\{\neg m_b\} & \text{if} \quad l = \neg m_b, a = s_m, \\
\{\neg m' \mid m' \in Cl(m), a = s_{m'}\} & \text{if} \quad l = \neg m_b, a \neq s_m.
\end{array}$$

The first three clauses of this definition are identical to the definition for the semantics without BCC. The difference is in the knowledge operators. Suppose $\varphi = \hat{K}_a\psi$. Again, I consider each literal in some member of $\mathcal{F}^n(\psi)$ separately.

If $l = m$ and $a \in \{s_m\} \cup r_m$ then m implies $\hat{K}_a m$ so I preserve m.

If $l = m$ and $a \notin \{s_m\} \cup r_m$ then a potentially knows m iff he sent or received some message in $\mathcal{M}^n_{Ag}(m)$, or if he was a BCC recipient of such a message.

If $l = \neg m$ then a potentially knows m iff there is some message in $Cl(m)$ of which he was the sender or a recipient which was not sent.

If $l = m_b$ or $l = \neg m_b$ and $a \in \{s_m\} \cup \{b\}$ then a certainly knows whether b was a BCC recipient of m so I preserve m_b or $\neg m_b$.

If $l = m_b$ and $a \notin \{s_m\} \cup \{b\}$ then a knows that b was a BCC recipient of m if a has received a forward (b, m, G) of m by b or a is the sender, recipient or BCC recipient of some message in $\mathcal{M}^n_{Ag}((b, m, G))$ for such a (b, m, G).

If $l = \neg m_b$ and $a \notin \{s_m\} \cup \{b\}$ then a knows b was not a BCC recipient of m if a knows that m was not sent, which is the case when some message in $Cl(m)$ of which a is a sender or a recipient was not sent.

For the case that $\varphi = \bar{K}_a\psi$, I also consider each literal separately.

If $l = m$ and $a = s_m$ I preserve m. If $a \neq s_m$ then a has definitive knowledge of m if he is the sender of some message in $\mathcal{M}^n_{Ag}(m)$.

If $l = \neg m$ then a has definitive knowledge of l if a is the sender of some message in $Cl(m)$ that was not sent.

If $l = m_b$ and $a = s_m$ then I preserve m_b. If $a \neq s_m$ then a definitively knows that b was a BCC recipient if he sent some message in $\mathcal{M}^n_{Ag}((b, m, G))$, for some group of agents G.

If $l = \neg m_b$ and $a = s_m$ then I preserve $\neg m_b$. If $a \neq s_m$ then a definitively knows b was not a BCC recipient of m if he definitively knows that m was not sent, which is the case if he was the sender of some message in $Cl(m)$ that was not sent.

I am convinced that the equivalent of Theorem 6.3.5 and 6.3.7 also hold for the case with BCC recipients.

6.5.6. Conjecture. *For any set of messages M and formula φ there is a finite number $n_{M,\varphi} \geq \delta(M)$ such that for every $k \geq n_{M,\varphi}$, $M \models \varphi$ iff any $F \in \mathcal{F}^k\varphi$ contains a literal $l \in F$ such that $M \models l$.*

6.5.7. Conjecture. *For any state M, formula φ and number $n > \delta(M)$, there is for every $F \in \mathcal{F}^n(\varphi)$ some $l \in F$ such that $M \models l$, if and only if the same holds for $\mathcal{F}^n(\varphi)|\delta(M)$.*

This would give a way to decide the semantics for the case with BCC recipients.

6.6 Conclusion

I have presented a logic that reasons about the knowledge of agents after a certain collection of messages or emails have been sent. Specifically I have focussed on the difference between having received a message and having replied to it. In the first case, it is not sure that the recipient has received the email in good order and also read it. In the second case it is. I have given a semantics based on the epistemic logic perspective, that is based on relations between states given by sets of messages or emails. The difference between messages and emails is that

the first only have a public list of recipients, while the second also have a secret list of BCC recipients.

Since the number of related states may be infinite, this perspective does not immediately give a way to decide the truth value of the formulas in finite time. Therefore I presented a way to decide each formula by looking at the truth value of certain literals. This decision procedure is proved correct for the case of messages. I also give a definition of this procedure for emails.

All in all I have presented a strong basis for a formal model checker that can be applied to sets of messages or emails in order to analyze who knows what in any situation where messages or emails are sent.

6.7 Proof of Theorem 6.3.5

I first state some facts that I will implicitly use throughout this section. I omit their proof, but they follow easily from the definition of closure and the semantics. For any two sets of messages M and N and any agent $a \in Ag$, the following hold:

- $Cl(Cl(M)) = Cl(M)$,
- $Cl(M \cup N) = Cl(M) \cup Cl(N)$,
- If $N \subseteq M$ then $Cl(N) \subseteq Cl(M)$,
- If $N \subseteq Cl(M)$ then $Cl(N) \subseteq Cl(M)$,
- $M \sim_a^P Cl(M)$ and $M \sim_a^D Cl(M)$,
- If $M \sim_a^P N$ and $M \models \hat{K}_a\varphi$ then $N \models \hat{K}_a\varphi$,
- If $M \sim_a^D N$ and $M \models \bar{K}_a\varphi$ then $N \models \bar{K}_a\varphi$.

6.7.1. Lemma. *For any set of messages M, $\Pi_a^*(M) \subseteq Cl(M)$. Similarly for Δ_a^*.*

Proof. Suppose $m \in \Pi_a^*(M) = Cl(\Pi_a(Cl(M)))$. Then there is $m' \in \Pi_a(Cl(M))$ that mentions m. Then $m' \in Cl(M)$, so because m' mentions m,

$$m \in Cl(Cl(M)) = Cl(M).$$

So $\Pi_a^*(M) \subseteq Cl(M)$. □

6.7.2. Lemma. *For any two sets of messages M and N, $M \sim_a^P N$ iff*

$$\Pi_a(Cl(M) \setminus Cl(N)) = \emptyset$$

and

$$\Pi_a(Cl(N) \setminus Cl(M)) = \emptyset.$$

Similarly for $\sim_a^D$ and Δ_a.

Proof. Take two sets of messages M and N and suppose $M \sim_a^P N$. For the sake of contradiction, suppose one of the sets mentioned above is non-empty. Without loss of generality, suppose there is some $m \in \Pi_a(Cl(M) \setminus Cl(N))$. Then $a \in \{s_m\} \cup r_m$ and $m \in Cl(M)$ and $m \notin Cl(N)$. Then $m \in \Pi_a(Cl(M))$ so $m \in \Pi_a^*(M)$. But because $M \sim_a^P N$, $\Pi_a^*(M) = \Pi_a^*(N)$ so then $m \in \Pi_a^*(N)$. But by Lemma 6.7.1 $\Pi_a^*(N) \subseteq Cl(N)$, so $m \in Cl(N)$. But I already knew that $m \notin Cl(N)$. This is a contradiction, so such m cannot exist and these sets must be empty.

For the converse I use contraposition. Suppose $M \not\sim_a^P N$. Then $\Pi_a^*(M) \neq \Pi_a^*(N)$. Without loss of generality, take $m \in \Pi_a^*(M) \setminus \Pi_a^*(N)$. Then there is $m' \in \Pi_a(Cl(M))$ that mentions m. Then $a \in \{s_{m'}\} \cup r_{m'}$ and $m' \in Cl(M)$. Suppose $m' \in Cl(N)$. Then $m' \in \Pi_a(Cl(N))$ so because m' mentions m, $m \in \Pi_a^*(N)$. This contradicts my assumption, so I conclude that $m' \notin Cl(N)$. So then $m' \in Cl(M) \setminus Cl(N)$. Then because $a \in \{s_{m'}\} \cup r_{m'}$, $m' \in \Pi_a(Cl(M) \setminus Cl(N))$. So $\Pi_a(Cl(M) \setminus Cl(N)) \neq \emptyset$. □

6.7.3. Lemma. *For any set of messages M and any message $m \in Cl(M)$, $M \models \hat{K}_a m$ iff $m \in \Pi_a^*(M)$. Similarly for $\bar{K}_a$ and Δ_a^*.*

Proof. Suppose $m \in \Pi_a^*(M)$. Then for any M' such that $M \sim_a^P M'$, $m \in \Pi_a^*(M') \subseteq Cl(M')$ so $M' \models m$. So $M \models \hat{K}_a m$. Conversely, suppose $M \models \hat{K}_a m$. Let $M' = Cl(M) \setminus \{m' \in Cl(M) \mid m' \text{ mentions } m\}$. Clearly, $M' \not\models m$ so $M \not\sim_a^P M'$. Note that $Cl(M') \setminus Cl(M) = \emptyset$ and $Cl(M) \setminus Cl(M') = \{m' \in Cl(M) \mid m' \text{ mentions } m\}$. So then by Lemma 6.7.2, there is $m' \in \Pi_a(Cl(M) \setminus Cl(M'))$. Then m' mentions m and $a \in \{s_{m'}\} \cup r_{m'}$. Then $m' \in \Pi_a^*(M)$ and $m \in \Pi_a^*(M)$. □

6.7.4. Lemma. *For any set of messages M and message m, either $M \models \hat{K}_a \neg m$ or $M \sim_a^P M \cup \{m\}$. Similarly for $\bar{K}_a$ and $\sim_a^D$.*

Proof. Suppose $M \not\sim_a^P M \cup \{m\}$. Then by Lemma 6.7.2 either $\Pi_a(Cl(M \cup \{m\}) \setminus Cl(M)) \neq \emptyset$ or $\Pi_a(Cl(M) \setminus Cl(M \cup \{m\})) \neq \emptyset$. Clearly, $Cl(M) \setminus Cl(M \cup \{m\}) = \emptyset$ so $\Pi_a(Cl(M) \setminus Cl(M \cup \{m\})) = \emptyset$. So I can take some $m' \in \Pi_a(Cl(M \cup \{m\}) \setminus Cl(M))$. Then $m' \in Cl(M \cup \{m\})$ and $m' \notin Cl(M)$. So $m' \in Cl(\{m\})$.

Take some M' such that $M \sim_a^P M'$. Suppose $m \in Cl(M')$. Then because $m' \in Cl(\{m\})$, $m' \in Cl(M')$ and because $a \in \{s_{m'}\} \cup r_{m'}$, $m' \in \Pi_a(Cl(M'))$. Then also $m' \in \Pi_a^*(M')$. But $M \sim_a^P M'$ so $\Pi_a^*(M') = \Pi_a^*(M)$ and $m' \in \Pi_a^*(M)$. But by Lemma 6.7.1 $\Pi_a^*(M) \subseteq Cl(M)$, so $m' \in Cl(M)$. But we already saw that $m' \notin Cl(M)$. This is a contradiction so $m \notin Cl(M')$ and $M' \not\models m$. But M' was chosen arbitrarily, so $M \models \hat{K}_a \neg m$.

The proof for $\bar{K}_a$ and $\sim_a^D$ is analogous. □

6.7.5. Lemma. *For any set of messages M and message m, either $M \models \hat{K}_a m$ or $M \sim_a^P Cl(M) \setminus \{m' \in Cl(M) \mid m' \text{ mentions } m\}$.*

Proof. Let $N = \{m' \in Cl(M) \mid m' \text{ mentions } m\}$. Suppose $M \not\sim_a^P Cl(M) \setminus N$. Then by Lemma 6.7.2 either $\Pi_a(Cl(M) \setminus Cl(Cl(M) \setminus N)) \neq \emptyset$ or $\Pi_a(Cl(Cl(M) \setminus N) \setminus Cl(M)) \neq \emptyset$. $Cl(M) \setminus N \subseteq Cl(M)$ so $Cl(Cl(M) \setminus N) \subseteq Cl(M)$ so

$\Pi_a(Cl(Cl(M)\backslash N)\backslash Cl(M)) = \emptyset$. So I can take some $m' \in \Pi_a(Cl(M)\backslash Cl(Cl(M)\backslash N))$. Then $a \in \{s_{m'}\} \cup r_{m'}$, $m' \in Cl(M)$ and $m' \notin Cl(Cl(M) \setminus N)$. Then $m' \notin Cl(M) \setminus N$, so because $m' \in Cl(M)$, $m' \in N$. So m' mentions m. Since $a \in \{s_{m'}\} \cup r_{m'}$ and $m' \in Cl(M)$, $m' \in \Pi_a(Cl(M))$. So $m \in \Pi_a^*(M)$. Take some M' such that $M \sim_a^P M'$. Then $\Pi_a^*(M) = \Pi_a^*(M')$ so $m \in \Pi_a^*(M')$. By Lemma 6.7.1 $\Pi_a^*(M') \subseteq Cl(M')$, so $m \in Cl(M')$ and $M' \models m$. But M' was chosen arbitrarily, so $M \models \hat{K}_a m$. The proof for Δ_a is analogous. □

6.7.6. Lemma. *Let $l_1, ..., l_n$ be literals such that $l_1 \vee ... \vee l_n$ is not a tautology. Let M be a set of messages such that $M \models \hat{K}_a(l_1 \vee ... \vee l_n)$. Then $M \models \hat{K}_a l_1 \vee ... \vee \hat{K}_a l_n$. Similarly for $\bar{K}_a$.*

Proof. I will give a proof with induction on the number of literals n. If $n = 1$ then the result becomes trivial. Suppose the result holds for n and take literals $l_1, ..., l_{n+1}$ and a set of messages M such that $M \models \hat{K}_a(l_1 \vee ... \vee l_{n+1})$. If $M \models \hat{K}_a(l_1 \vee ... \vee l_n)$ then the result follows by induction hypothesis. Suppose otherwise. Then there is some M' such that $M \sim_a^P M'$ and $M' \models \neg l_1 \wedge ... \wedge \neg l_n$. Then because $M \models \hat{K}_a(l_1 \vee ... \vee l_{n+1})$, it must be the case that $M' \models l_{n+1}$. We claim that $M \models \hat{K}_a l_{n+1}$. Suppose otherwise.

Suppose $l_{n+1} = m$ for some message m. Let $N = \{m' \in Cl(M') \mid m' \text{ mentions } m\}$. By Lemma 6.7.5, the fact that $M' \not\models \hat{K}_a l_{n+1}$ implies that $M' \sim_a^P Cl(M') \setminus N$. Clearly, $Cl(M')\backslash N \not\models l_{n+1}$. Suppose there is some l_a such that $Cl(M')\backslash N \models l_a$. I already know that $M' \not\models l_a$, so then it must be the case that $l_a = \neg m'$ and $m' \in N$. But then m' mentions m and $l_1 \vee ... \vee l_n$ is a tautology. So $Cl(M') \setminus N \not\models l_a$ for any l_a. But I assumed that $M \models \hat{K}_a(l_1 \vee ... \vee l_{n+1})$, so this is a contradiction.

Suppose $l_{n+1} = \neg m$ for some message m. By Lemma 6.7.4 the fact that $M' \not\models \hat{K}_a l_{n+1}$ implies that $M' \sim_a^P M' \cup \{m\}$. Clearly, $M' \cup \{m\} \not\models l_{n+1}$. Suppose there is some l_a such that $M' \cup \{m\} \models l_a$. I already know that $M' \not\models l_a$ so then it must be the case that $l_a = m'$ for some message $m' \in Cl(m)$. But then $l_1 \vee ... \vee l_n$ is a tautology. So $M' \cup \{m\} \not\models l_a$ for any l_a. But I assumed that $M \models \hat{K}_a(l_1 \vee ... \vee l_{n+1})$, so this is a contradiction.

I conclude that $M \models \hat{K}_a l_{n+1}$. The proof for $\bar{K}_a$ is analogous. □

6.7.7. Lemma. *Let M, M' be sets of messages and let $l_1, ..., l_n$ be literals such that $M \sim_a^P M'$ and $M' \models l_1 \wedge ... \wedge l_n$. Then there is M'' such that $M \sim_a^P M''$, $M'' \models l_1 \wedge ... \wedge l_n$ and $\delta(M'') \leq \max(\delta(M), \delta(l_1), ..., \delta(l_n))$. Similarly for $\sim_a^D$.*

Proof. First note that because $M \sim_a^P M'$ and $M' \models l_1 \wedge ... \wedge l_n$, for any l_a I have that $M \not\models \hat{K}_a \neg l_a$. Let $M^+ = \{m \in \{l_1, ..., l_n\} \mid M \not\models m\}$. For any $m \in M^+$, $M \not\models \hat{K}_a \neg m$ so then by repeated application of Lemma 6.7.4 I get that $M \sim_a^P M \cup M^+$. Let $M^- = \{m \in Cl(M \cup M^+) \mid m \text{ mentions some } m' \text{ such that } \neg m' \in$

$\{l_1, ..., l_n\}\}$. For any $\neg m' \in \{l_1, ..., l_n\}$ it holds that $M \not\models \hat{K}_a m'$ so then $M \cup M^+ \not\models \hat{K}_a m'$. Then by repeated application of Lemma 6.7.4 I get that $M \sim_a^P Cl(M \cup M^+) \setminus M^-$. Clearly, every l_a of the form $l_a = \neg m$ is satisfied in $Cl(M \cup M^+) \setminus M^-$. Now take some l_a of the form $l_a = m$. Clearly, $m \in Cl(M \cup M^+)$. Suppose $Cl(M \cup M^+) \setminus M^- \not\models m$. Then $m \in M^-$, so m mentions some m' such that $\neg m' \in \{l_1, ..., l_n\}$. But then $l_1 \wedge ... \wedge l_n$ is a contradiction which is not possible because $M' \models l_1 \wedge ... \wedge l_n$. So $Cl(M \cup M^+) \setminus M^- \models l_1 \wedge ... \wedge l_n$. It is not hard to see that $\delta(Cl(M \cup M^+) \setminus M^-) \leq \max(\delta(M), \delta(l_1), ..., \delta(l_n))$. □

6.7.8. Corollary. *Let M be a set of messages and $l_1, ..., l_n$ be literals. Suppose that for any $M' \sim_a^P M$ with $\delta(M') \leq \max(\delta(M), \delta(l_1), ..., \delta(l_n))$, $M' \models l_1 \vee ... \vee l_n$. Then for any M'' such that $M \sim_a^P M''$, $M'' \models l_1 \vee ... \vee l_n$.*

6.7.9. Theorem. *For any set of messages M and formula φ there is a finite number $n_{M,\varphi} \geq \delta(M)$ such that for every $k \geq n_{M,\varphi}$,*

$$M \models \varphi \text{ iff any } F \in \mathcal{F}^k\varphi \text{ contains is a literal } l \in F \text{ such that } M \models l.$$

Proof. I will give a proof with structural induction on φ.
Suppose $\varphi = m$. Let $n_{M,\varphi} = \max(\delta(M), \delta(m))$. Then for any $k \geq n_{M,\varphi}$, $\mathcal{F}^k(\varphi) = \{\{m\}\}$ and the desired result follows immediately.
Suppose $\varphi = \neg\psi$. Let $n_{M,\varphi} = n_{M,\psi}$ and take some $k \geq n_{M,\varphi}$. Suppose $M \models \neg\psi$. Then there is F in $\mathcal{F}^k(\psi)$ such that for every $l \in F$, $M \models \bar{l}$. Then for every $F' \in \mathcal{F}^k(\neg\psi)$ there is $\bar{l} \in F'$ such that $l \in F$ and $M \models \bar{l}$. For the converse I will use contraposition. Suppose that $M \models \psi$. Then for every $F \in \mathcal{F}^k(\psi)$ there is some $l \in F$ such that $M \models l$. Let $F' \in \mathcal{F}^k(\neg\psi)$ be the set containing the negation of exactly these literals. Then there is no $\bar{l} \in F'$ such that $M \models \bar{l}$. So then it does not hold that every $F' \in \mathcal{F}^k(\neg\psi)$ contains some $l' \in F'$ such that $M \models l'$.
Suppose $\varphi = \psi_1 \wedge \psi_2$. Let $n_{M,\varphi} = \max(n_{M,\psi_1}, n_{M,\psi_2})$. The result follows by definition and induction hypothesis.
Suppose $\varphi = \hat{K}_a\psi$. Construct $n_{M,\varphi}$ as follows. If $M \models \hat{K}_a\psi$ then $n_{M,\varphi} = \max(\delta(\psi), n_{M,\psi})$. Otherwise, let k_1 be the minimal number such that $k_1 = n_{M_1,\psi}$ for some state M_1 such that $M_1 \models \neg\psi$ and $M \sim_a^P M_1$. Let $n_{M,\varphi} = \max(\delta(\psi), n_{M,\psi}, k_1)$. Take some $k \geq n_{M,\varphi}$.

Suppose $M \models \hat{K}_a\psi$. Take some $F \in \mathcal{F}^k(\hat{K}_a\psi)$. Then there is some $F' \in \mathcal{F}^k(\psi)$ on which F is based. Suppose $F' = \{l_1, ..., l_n\}$. Let $\mathbf{M}$ be the collection of sets of messages M' such that $M \sim_a^P M'$ and $\delta(M') \leq k$. This collection is finite. For any $M' \in \mathbf{M}$, $M' \models \psi$ and by induction hypothesis, $M' \models l_1 \vee ... \vee l_n$. Note that $\max(\delta(M), \delta(l_1), ..., \delta(l_n)) \leq k$. So by Corollary 6.7.8, $M \models \hat{K}_a(l_1 \vee ... \vee l_n)$.

Then by Lemma 6.7.6, $M \models \hat{K}_a l_1 \vee ... \vee PK_a l_n$. Take some l_j such that $M \models \hat{K}_a l_j$. I claim that $M \models l$ for some $l \in F$ based on l_j.

Suppose $l_j = m$ and $a \in \{s_m\} \cup r_m$. Then let $l = m$ and I am done.
Suppose $l_j = m$ and $a \notin \{s_m\} \cup r_m$. Because $M \models \hat{K}_a m$, I have by Lemma 6.7.3 that $m \in \Pi_a^*(M)$. So there must be some $m'' \in \Pi_a^(Cl(M))$ mentioning m. Then $a \in \{s_{m''}\} \cup r_{m''}$ so $m'' \neq m$ and there must be b, G such that $m'' = (b, m', G)$, where m' mentions m and $a \in \{b\} \cup G$. Also, $m'' \in Cl(M)$ so $M \models m''$. Clearly, $m'' \in F'$. I let $l = m''$ and I am done.
Suppose $l_j = \neg m$. Let $M' = M \cup \{m\}$. Then because $M \models \hat{K}_a \neg m$, $M \not\sim_a M'$. Note that $Cl(M) \setminus Cl(M') = \emptyset$ so then by Lemma 6.7.2 there is $m' \in \Pi_a(Cl(M') \setminus Cl(M))$. But if $m' \in Cl(M \cup \{m\}) \setminus Cl(M)$ then $m' \in Cl(\{m\})$. So m' is mentioned in m. Also, if $m' \in \Pi_a(Cl(M') \setminus Cl(M))$ then $a \in \{s_{m'}\} \cup r_{m'}$ so $\neg m' \in F'$. But $m' \notin Cl(M)$ so $M \models \neg m'$.

Since F was chosen arbitrarily from $\mathcal{F}^k(\hat{K}_a \psi)$, this proves the desired result.

Now, suppose that for any $F \in \mathcal{F}^k(\hat{K}_a \psi)$, there is $l \in F$ such that $M \models l$. For the sake of contradiction, suppose $M \not\models K_a \psi$. Then by construction of $n_{M,\varphi}$ there is some M_1 such that $M_1 \models \neg\psi$, $M \sim_a^P M_1$ and $n_{M_1,\psi} \leq n_{M,\varphi}$.

I claim that for any $F' \in \mathcal{F}^k(\psi)$, there is $l' \in F'$ such that $M_1 \models l'$. Take such F'. Let $F \in \mathcal{F}^k(\varphi)$ be the set based on F' and take $l \in F$ such that $M \models l$. Let $l' \in F'$ be the literal on which l is based. I claim that $M_1 \models l'$.
Suppose $l = l' = m$ and $a \in \{s_m\} \cup r_m$. Then $m \in \Pi_a^*(M)$ and by Lemma 6.7.3, $M \models \hat{K}_a m$ so $M_1 \models m$.
Suppose $l = (j, m', G)$, $l' = m$, m' mentions m, $a \notin \{s_m\} \cup r_m$ and $a \in \{b\} \cup G$. Then $(b, m', G) \in \Pi_a^*(M)$, so again by Lemma 6.7.3 $M_1 \models (b, m', G)$. But since $m \in Cl(m')$ and $m' \in Cl((b, m', G))$, then $M_1 \models m$.
Suppose $l = \neg m'$, $l' = \neg m$, m mentions m' and $a \in \{s_{m'}\} \cup r_{m'}$. For the sake of contradiction suppose $M_1 \models m$. Then because $m' \in Cl(m)$, $M_1 \models m'$. But $a \in \{s_{m'}\} \cup r_{m'}$ so then $m' \in \Pi_a^*(M_1)$ and by Lemma 6.7.3 $M \models m'$. But this contradicts my assumption that $M \models l$. So then it must be the case that $M_1 \models \neg m$.

Suppose $n_{M_1,\psi} \leq k$. Then we can apply the induction hypothesis to derive that $M_1 \models \psi$, which is a contradiction with my earlier claim. So $n_{M_1,\psi} > k \geq n_{M,\varphi}$. But this contradicts the construction of M_1. We conclude that our assumption that $M \not\models \hat{K}_a \psi$ was false, so $M \models \hat{K}_a \psi$.
Suppose $\varphi = \bar{K}_a \psi$. The proof is analogous to that for $\hat{K}_a \psi$. □

Chapter 7

Action Emulation

7.1 Introduction

In this thesis I often use Kripke models to model the knowledge of a group of agents in a certain situation. In Chapters 3 and 8 I also use action models to update these models when the situation changes. In this chapter I will address an important technical question concerning these models, namely: when are two action models equivalent? And how can one detect such an equivalence?

Kripke models may be used to interpret any modal logic and they are well studied. In particular, it is well known (see e.g. [Blackburn et al., 2001]) that two Kripke models are semantically equivalent if and only if there exists a relation between them that is a bisimulation.

Action models were introduced in [Baltag et al., 1998] as a way to model communicative actions rather than static situations. Two action models are considered equivalent if they have the same effect on all possible Kripke models. However, up to now there is no notion corresponding to bisimulation for action models. In other words, there is no easy way to tell whether two action models are equivalent just by looking at their structure. This chapter is dedicated to finding the right definition of a relation between action models called *action emulation*, such that there exists an action emulation between two action models if and only if they are equivalent.

The problem I study here has been addressed before in [van Eijck et al., 2012]. There, a partial solution is provided. A notion of action emulation parameterized by the worlds of a canonical Kripke model is constructed. The union of all these relations is shown to coincide with action model equivalence. This is a step forward, but not the final word. Using this notion of action emulation one would have to construct a relation between the action models for every world from a canonical Kripke model, which is tedious work. I would like to improve on this result by giving a direct definition of action emulation between action models. The definition I propose here is a lot simpler than the one from [van

Eijck et al., 2012] because it does not involve worlds from a canonical Kripke model and is constructed as one single relation, rather than being the union of multiple relations. This is an advantage because the canonical Kripke model has a great number of worlds and computing a relation for each of these worlds takes a lot of time.

This chapter is set up as follows. First I give some established definitions related to Kripke models and action models. Then I introduce the class of canonical action models and show that every action model has an equivalent canonical action model. I give a definition of action emulation and show that the existence of an action emulation between two action models implies their equivalence. Then I prove that the converse holds for the class of canonical action models. Because any action model has an equivalent canonical action model, this way any two action models can be compared.

7.2 Definitions

Let P be a countable set of proposition letters an let A be a finite set of action labels. The modal language $\mathcal{L}^M$ over P and A is given by:

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \Diamond_a \phi$$

where p ranges over P and a over A. This is very similar to the language of DEL presented in Chapter 2, only instead of epistemic programs I use a modality, $\Diamond_a\phi$. It may stand for knowledge, obligation, or any other of a wide range of interpretations.

I will use the usual shorthands: $\phi \wedge \psi$ for $\neg(\phi \vee \psi)$, $\phi \to \psi$ for $\neg\phi \vee \psi$ and $\Box_a\phi$ for $\neg(\Diamond_a\neg\phi)$. The modality $\Box_a\phi$ is the dual of $\Diamond_a\phi$.

Given a formula φ, I define its ***single negation*** as follows: if ϕ is of the form $\neg\psi$, then $\sim\phi = \psi$, and otherwise $\sim\phi = \neg\phi$. I will implicitly use the equivalences of $\neg\Box_a\phi$ and $\Diamond_a\sim\phi$, of $\neg\Diamond_a\phi$ and $\Box_a\sim\phi$, of $\neg(\phi\wedge\psi)$ and $\sim\phi\vee\sim\psi$, and of $\neg(\phi\vee\psi)$ and $\sim\phi \wedge \sim\psi$.

The definition of single negation allows me to define the closure of a formula or a set of formulas.

7.2.1. Definition. Given a formula ϕ, I define its closure $C(\phi)$ as the smallest set containing ϕ that is closed under taking subformulas and single negations. Given a finite set of formulas Φ, I define $C(\Phi) := \bigcup_{\phi\in\Phi} C(\phi)$.

The following example shows how this definition works out.

7.2.2. Example. $p \wedge \Diamond_a\neg p$ has the following closure:

$$C(p \wedge \Diamond_a\neg p) = \{p \wedge \Diamond_a\neg p, \neg p \vee \Box_a p, p, \neg p, \Diamond_a\neg p, \Box_a p\}.$$

7.2.3. Definition. An atom over a finite set of formulas Φ is a maximal subset of $C(\Phi)$ which is consistent (in the K axiomatisation of multi-modal logic).

An atom over Φ can be seen as a complete description of a possible state of the world, if one only considers the formulas in Φ. I will use these atoms later on to construct canonical models.

7.2.4. Example. $\{p \wedge \Diamond_a \neg p\}$ has four atoms:

- $\{p \wedge \Diamond_a \neg p, p, \Diamond_a \neg p\}$,
- $\{\neg p \vee \Box_a p, \neg p, \Diamond_a \neg p\}$,
- $\{\neg p \vee \Box_a p, p, \Box_a p\}$,
- $\{\neg p \vee \Box_a p, \neg p, \Box_a p\}$.

I will interpret the formulas from $\mathcal{L}^M$ on Kripke models. These are defined in Chapter 2. I will use a set of action labels A instead of a set of agents. This is because the modalities $\Diamond_a$ and $\Box_a$ do not necessarily represent the knowledge of an agent.

In Chapter 2, the relations of a Kripke model were assumed to be reflexive, symmetric and transitive. Here, I no longer make this assumption. Therefore instead of using $\sim_a$ as an alternate notation for R_a, I will now use $\xrightarrow{a}$.

The semantics of $\mathcal{L}^M$ is mostly as defined in Chapter 2. A formal definition is as follows:

$$\begin{array}{rll} \mathcal{M} \models_w p & \text{iff} & p \in \mathit{Val}(w) \\ \mathcal{M} \models_w \neg\phi & \text{iff} & \mathcal{M} \not\models_w \phi \\ \mathcal{M} \models_w \phi_1 \vee \phi_2 & \text{iff} & \mathcal{M} \models_w \phi_1 \text{ or } \mathcal{M} \models_w \phi_2 \\ \mathcal{M} \models_w \Diamond_a \phi & \text{iff} & \exists w' : w R_a w' \text{ and } w' \models \phi. \end{array}$$

The semantics of the modality $\Diamond_a$ is straightforward: $\Diamond_a \phi$ holds if it is possible to do an a-step to a world where ϕ holds. Dually, $\Box_a \phi$ holds if every world that is reachable with an a-step satisfies ϕ.

7.3 Bisimilar Action Models

As discussed in Chapter 2, two Kripke models are considered equivalent when they are bisimilar. If they are bisimilar, they satisfy exactly the same modal formulas. They can be considered two different models of the exact same situation.

Action models model a communicative event. Just like Kripke models, sometimes two different action models model the same thing. In the case of action models, this means they model the same communicative event. This is signified by the fact that they have the same effect on all Kripke models. That is, if the two different action models are applied to the same Kripke model, the resulting models will be bisimilar.

7.3.1. Definition. Take two action models $\mathcal{A}$ and $\mathcal{B}$ over a set of agents Ag and a set of propositions P. I will say that $\mathcal{A}$ and $\mathcal{B}$ are equivalent, notation $\mathcal{A} \equiv \mathcal{B}$, if for any Kripke model $\mathcal{M}$ over Ag and Q, where $P \subseteq Q$,

$$\mathcal{M} \otimes \mathcal{A} \leftrightarrows \mathcal{M} \otimes \mathcal{B}.$$

Note that if two action models are equivalent, then the result of updating a Kripke model with one of them is bisimilar to the result of the update with the other, even if the model mentions propositions that are not mentioned in the action models. Usually, I will apply action models over a certain set of propositions to Kripke models over the same set of propositions. However, in Lemma 7.3.7 I will make use of the fact that equivalence still holds when the Kripke model has propositions that are not mentioned in the action model.

The problem I face in this chapter is to find a structural relation between action models that signifies their equivalence, just like bisimulation does for Kripke models. When two action models $\mathcal{A}$ and $\mathcal{B}$ are equivalent, every world that matches an event of $\mathcal{A}$ should also match an event of $\mathcal{B}$ and vice versa. Furthermore, the results of these matchings should be bisimilar.

The first solution that comes to mind is to apply bisimulation to action models. One could replace the requirement that the worlds have the same valuation with the requirement that their preconditions are semantically equivalent. This gives the following definition:

7.3.2. Definition. Two action models $\mathcal{A}$ and $\mathcal{B}$ are bisimilar if there is a relation $Z : E^{\mathcal{A}} \times E^{\mathcal{B}}$ which is total on $E_0^{\mathcal{A}} \times E_0^{\mathcal{B}}$, such that the following conditions hold for any x, y such that xZy:

Invariance $Pre^{\mathcal{A}}(x) \equiv Pre^{\mathcal{B}}(y)$,

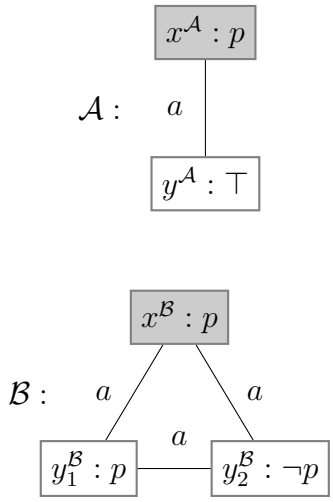
Zig for any action label $a \in A$, if there is a world x' such that $x \xrightarrow{a}{}^{\mathcal{A}} x'$ then there must be a world y' such that $y \xrightarrow{a}{}^{\mathcal{B}} y'$ and $(x', y') \in Z$,

Zag for any action label $a \in A$, if there is a world y' such that $y \xrightarrow{a}{}^{\mathcal{B}} y'$ then there must be a world x' such that $x \xrightarrow{a}{}^{\mathcal{A}} x'$ and $(x', y') \in Z$.

Here $\equiv$ signifies logical equivalence.

However, this bisimulation for action models does not have the required properties. The following example, which is inspired by [van Eijck et al., 2012], shows why not.

7.3.3. Example. Consider the following two action models, where all relations are symmetric, and reflexive relations are present for all events but not drawn in the picture.



These two models are not bisimilar: there is no event in $\mathcal{B}$ that has a precondition which is logically equivalent to the precondition of $y^{\mathcal{A}}$ in $\mathcal{A}$. Therefore the a-step from the actual world $x^{\mathcal{A}}$ to $y^{\mathcal{A}}$ cannot be matched by an a-step from $x^{\mathcal{B}}$ to a world that is bisimilar to $y^{\mathcal{A}}$.

However, they are equivalent. One can see this as follows. Clearly any world that matches event $x^{\mathcal{A}}$ in $\mathcal{A}$ will match event $x^{\mathcal{B}}$ in $\mathcal{B}$ and vice versa. Furthermore, any world that matches event $y^{\mathcal{A}}$ in $\mathcal{A}$ will match $y_1^{\mathcal{B}}$ in $\mathcal{B}$ if it satisfies p, and $y_2^{\mathcal{B}}$ in $\mathcal{B}$ if it does not satisfy p. Since the relations between $x^{\mathcal{B}}$ and $y_1^{\mathcal{B}}$ and $y_2^{\mathcal{B}}$ in $\mathcal{B}$ are the same as the relations between $x^{\mathcal{A}}$ and $y^{\mathcal{A}}$ in $\mathcal{A}$, the results of these matchings are bisimilar.

More formally, if $\mathcal{M}$ is a Kripke model then I define the relation Z on $W^{\mathcal{M}\otimes\mathcal{A}} \times W^{\mathcal{M}\otimes\mathcal{B}}$ as follows. For any $w \in W^{\mathcal{M}}$,

$$\begin{array}{ll} (w, x^{\mathcal{A}}) Z (w, x^{\mathcal{B}}), & \\ (w, y^{\mathcal{A}}) Z (w, y_1^{\mathcal{B}}) & \text{if } w \models p, \\ (w, y^{\mathcal{A}}) Z (w, y_2^{\mathcal{B}}) & \text{otherwise.} \end{array}$$

It is not hard to check that Z is indeed a bisimulation between $\mathcal{M}\otimes\mathcal{A}$ and $\mathcal{M}\otimes\mathcal{B}$.

The above example shows that the problem of detecting equivalence between action models is not solved by simply adapting the definition of bisimulation. Therefore I would like to find a more sophisticated relation between action models. I will define such a relation later in this chapter, but first I will show that there is a way to detect action model equivalence by looking at canonical Kripke models.

A canonical Kripke model is a model that has a world for every possible atom over a certain set of formulas. It models all possible truth values of these formulas and their subformulas.

7.3.4. Definition. If Φ is a finite set of formulas and Σ the set of atoms over Φ, then the canonical Kripke model $\mathcal{M}^c = (W^c, Val^c, R^c, W_0^c)$ over Φ is defined

as

$$\begin{array}{lcl} W^c & := & \Sigma \\ Val^c(\sigma) & := & P \cap \sigma \\ \sigma \stackrel{a}{\rightarrow}{}^c \sigma' & \text{iff} & \bigwedge \sigma \wedge \Diamond_a \bigwedge \sigma' \text{ is consistent} \\ W_0^c & := & \Sigma \end{array}$$

Every world in the canonical model corresponds to an atom, and there is an a-relation from one atom to another if the formulas in the first atom are consistent with $\Diamond_a\phi$, for any formula ϕ in the second atom. The following is shown in [Blackburn et al., 2001].

7.3.5. Theorem. *Let $\mathcal{M}^c$ be the canonical model over a set of formulas Φ. Then for any atom σ over Φ and for any formula $\phi \in C(\Phi)$,*

$$\mathcal{M}^c \models_\sigma \phi \text{ iff } \phi \in \sigma.$$

Given an action model $\mathcal{A}$, I define its ***language*** $\Lambda_\mathcal{A}$ as the closure of the union of the preconditions of all its events. In [van Eijck et al., 2012], the following very useful observation is made about canonical Kripke models and action model equivalence:

7.3.6. Theorem. *Take two action models $\mathcal{A}$ and $\mathcal{B}$ such that $\Phi = \Lambda_\mathcal{A} \cup \Lambda_\mathcal{B}$ and let $\mathcal{M}^c$ be the canonical Kripke model over Φ. Then the following holds:*

$$\mathcal{A} \equiv \mathcal{B} \text{ iff } \mathcal{M}^c \otimes \mathcal{A} \leftrightarrows \mathcal{M}^c \otimes \mathcal{B}.$$

A proof of this theorem is given in [van Eijck et al., 2012]. However, the proof given there is slightly lacking: it makes an assumption that is not properly shown to be true. In order to be entirely correct, the proof would need to be preceded by the following lemma. It states that if two action models $\mathcal{A}$ and $\mathcal{B}$ are equivalent and they are applied to some epistemic model $\mathcal{M}$ then one can find not only a bisimulation between $\mathcal{M} \otimes \mathcal{A}$ and $\mathcal{M} \otimes \mathcal{B}$, but also one that connects only pairs that result from the same world in $W^\mathcal{M}$.

7.3.7. Lemma. *Take two action models $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A} \equiv \mathcal{B}$. Then for any model $\mathcal{M}$ of countable size there is a bisimulation Z between $\mathcal{M} \otimes \mathcal{A}$ and $\mathcal{M} \otimes \mathcal{B}$ such that $(w, x)Z(v, y)$ implies $w = v$.*

Proof. Take some model $\mathcal{M}$. Let P be the set of propositions. For every world $w \in W^\mathcal{M}$ construct a new proposition p_w which is not in P. Let $\mathcal{M}'$ be a model over $P \cup \{p_w \mid w \in W_\mathcal{M}\}$ which is identical to $\mathcal{M}$, except for the fact that the valuation is extended in such a way that every new proposition p_w is true in world w and false in all other worlds. Because $\mathcal{A} \equiv \mathcal{B}$, there must be a bisimulation Z between $\mathcal{M}' \otimes \mathcal{A}$ and $\mathcal{M}' \otimes \mathcal{B}$. Because every world in $W_{\mathcal{M}'}$ has a unique valuation

that is preserved in the action update, it holds that $(w, x)Z(v, y)$ implies $w = v$. But clearly, Z is also a bisimulation between $\mathcal{M} \otimes \mathcal{A}$ and $\mathcal{M} \otimes \mathcal{B}$. So I have shown that for any model $\mathcal{M}$ there exists such a bisimulation with the desired property. □

Using this lemma, the proof of Theorem 7.3.6 goes as follows. This follows [van Eijck et al., 2012] almost precisely, except for the fact that there the existence of a bisimulation as constructed in Lemma 7.3.7 is not proven.

Proof of Theorem 7.3.6. The proof for the left to right direction is immediate by the definition of action model equivalence. For the right to left direction, suppose $\mathcal{M}^c \otimes \mathcal{A} \leftrightarrows \mathcal{M}^c \otimes \mathcal{B}$. Then by Lemma 7.3.7 there is a bisimulation $Z : W^{\mathcal{M}^c \otimes \mathcal{A}} \times W^{\mathcal{M}^c \otimes \mathcal{B}}$ with the special property that $(w, x)Z(v, y)$ implies $w = v$. Take any Kripke model $\mathcal{M}$. Define a relation $Y : W^{\mathcal{M} \otimes \mathcal{A}} \times W^{\mathcal{M} \otimes \mathcal{B}}$ as follows:

$$(w, x)Y(v, y) \text{ iff } w = v \text{ and } (w^*, x)Z(w^*, y),$$

where given some $w \in W^{\mathcal{M}}$, $w^* \in W^{\mathcal{M}^c}$ is defined as the atom that consists of all elements of $C(\Phi)$ that are satisfied in w. I will show that Y is a bisimulation. Suppose $(w, x)Y(w, y)$. Then $(w^*, x)Z(w^*, y)$.

To see that Invariance is satisfied, observe that the valuations of (w, x) and of (w, y) are both inherited from w and therefore identical.

For Zig, suppose $(w, x) \xrightarrow{a} (w', x')$. Then $w \xrightarrow{a} w'$ and $x \xrightarrow{a} x'$. Because $\mathcal{M} \models_w \bigwedge w^*$ and $\mathcal{M} \models_{w'} \bigwedge w'^*$, then $\bigwedge w^* \wedge \Diamond_a(\bigwedge w'^*)$ is consistent, so $w^* \xrightarrow{a} w'^*$. Because $\mathcal{M} \models_{w'} \mathit{Pre}(x')$, it holds that $\mathit{Pre}(x') \in w'^*$. So $(w^*, x) \xrightarrow{a} (w'^*, x')$. But since $(w^*, x)Z(w^*, y)$, then there must be (v, y') such that $(w^*, y) \xrightarrow{a} (v, y')$ and $(w'^*, x')Z(v, y')$. Then by the special property of Z I have $v = w'^*$, so $(w'^*, x')Z(w'^*, y')$. So $(w', x')Y(w', y')$. Since $(w^*, y) \xrightarrow{a} (w'^*, y')$ it holds that $y \xrightarrow{a} y'$. Since I already knew that $w \xrightarrow{a} w'$, this shows $(w, y) \xrightarrow{a} (w', y')$.

The proof for Zag is analogous.

To see that Y is total, take some $(w, x) \in W_0^{\mathcal{M} \otimes \mathcal{A}}$. Then $\mathcal{M} \models_w \mathit{Pre}(x)$ so $\mathit{Pre}(x) \in w^*$. Then $\mathcal{M}^c \models_{w^*} \mathit{Pre}(x)$, so $(w^*, x) \in \mathcal{M}^c \otimes \mathcal{A}$. Then by the special property of Z there is some $y \in \mathcal{B}$ such that $(w^*, x)Z(w^*, y)$. So $\mathcal{M}^c \models_{w^*} \mathit{Pre}(y)$, and then $\mathit{Pre}(y) \in w^*$ so $\mathcal{M} \models_w \mathit{Pre}(y)$. So $(w, x)Y(w, y)$. □

This theorem demonstrates a straightforward procedure to check whether two action models are equivalent: simply construct the canonical Kripke model for the set of formulas consisting of their preconditions, and see whether the update results on this model bisimulate. Even though this is not complicated, it is a very inefficient method: the size of the canonical Kripke model is exponential in the number of subformulas of the preconditions.

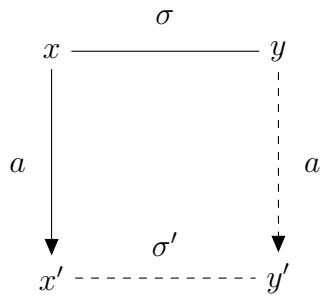
I am looking for a definition of a direct relation between action models that signifies their equivalence. Inspired by the above theorem, in [van Eijck et al.,

2012] a relation is constructed which is parameterized by worlds in the canonical Kripke model. This parameterized action emulation does not yet lead to an efficient method, because every world in the canonical Kripke model has to bee computed. However, I take it as a starting point for further investigations. It is defined as follows.

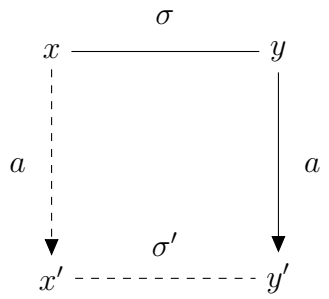
7.3.8. Definition. Given two action models $\mathcal{A}$ and $\mathcal{B}$, let Σ be the set of atoms over $\Lambda_{\mathcal{A}} \cup \Lambda_{\mathcal{B}}$. Given some $x \in E^{\mathcal{A}} \cup E^{\mathcal{B}}$, let $S(x) = \{\sigma \in \Sigma \mid \mathit{Pre}(x) \in \sigma\}$. An action emulation between $\mathcal{A}$ and $\mathcal{B}$ is a set of indexed relations $\{E_\sigma\}_{\sigma\in\Sigma}$ such that whenever $xE_\sigma y$ the following conditions hold:

Invariance $\mathit{Pre}(x) \in \sigma$ and $\mathit{Pre}(y) \in \sigma$.

Zig If $x \xrightarrow{a} x'$ then for any $\sigma' \in S(x')$ such that $\sigma \xrightarrow{a} \sigma'$ there is $y' \in E^{\mathcal{B}}$ with $y \xrightarrow{a} y'$ and $x'E_{\sigma'}y'$. In a picture:



Zag If $y \xrightarrow{a} y'$ then for any $\sigma' \in S(y')$ such that $\sigma \xrightarrow{a} \sigma'$ there is $x' \in E^{\mathcal{A}}$ with $x \xrightarrow{a} x'$ and $x'E_{\sigma'}y'$. In a picture:



I say that $\mathcal{A}$ and $\mathcal{B}$ emulate parameterized by the canonical model if for every $x \in E_0^{\mathcal{A}}$ and for every $\sigma \in S(x)$ there is $y \in E_0^{\mathcal{B}}$ with $xE_\sigma y$, and vice versa. Notation: $\mathcal{A} \leftrightarrows_S \mathcal{B}$.

It is shown in [van Eijck et al., 2012] that this relation indeed characterizes action model equivalence:

7.3.9. THEOREM. *For any two action models $\mathcal{A}$ and $\mathcal{B}$,*

$$\mathcal{A} \equiv \mathcal{B} \textit{ iff } \mathcal{A} \leftrightarrows_S \mathcal{B}.$$

To see why this definition works, observe that any world w from any Kripke model $\mathcal{M}$ has a corresponding atom w^*. Then if $\mathcal{A} \leftrightarrows_S \mathcal{B}$, there must be for every $x \in E_\mathcal{A}$ such that $\mathcal{M} \models_w Pre(x)$ some event $y \in E_\mathcal{B}$ such that $xE_{w^*}y$. Then $\mathcal{M} \models_w Pre(y)$, and it is not hard to show that (w, x) is bisimilar to (w, y).

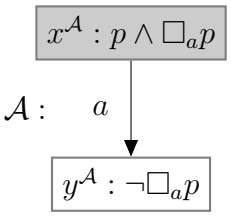
However, this definition leaves me with the same problem as before: it requires the computation of a large number of atoms. One even has to compute a separate relation for every possible atom! This is very inefficient. Therefore I want to improve on this by finding a non-parameterized notion of action emulation.

Checking whether two action models are equivalent is complicated because one world from a Kripke model may match multiple events in the action model and one event in the action model may match multiple worlds in the Kripke model. Moreover, usually there is no direct mapping between $\mathcal{A}$ and $\mathcal{B}$ such that an event in $\mathcal{A}$ matches the exact same worlds in the Kripke model as the related event in $\mathcal{B}$. To circumvent these complications I consider canonical action models.

7.3.10. DEFINITION. An action model $\mathcal{A}$ is canonical over a finite set of $\mathcal{L}^M$ formulas Φ if every precondition is the conjunction of an atom over Φ and for every $x, x' \in E_A$ such that $x \xrightarrow{a} x'$, $Pre(x) \wedge \Diamond_a Pre(x')$ is consistent.

Note the difference between canonical Kripke models and canonical action models: a canonical Kripke model has a world for every possible atom, and has a relation between two worlds if and only if this relation is consistent with the contents of the atoms. On the other hand, a canonical action model may be incomplete in the sense that there may be atoms that are not represented as the precondition of an event in the model. Also, a relation between two events may not be present even though it would be consistent with the preconditions of the events.

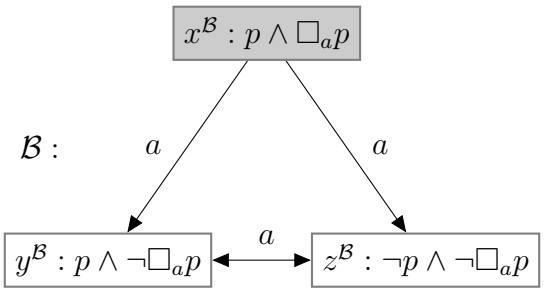
7.3.11. EXAMPLE. Consider the following action model (reflexive relations present but omitted in the picture):



This action model is not canonical. The reason for this is that the precondition of world $y^\mathcal{A}$ is not the conjunction of an atom over the set of formulas $\{p, \Box_a p\}$.

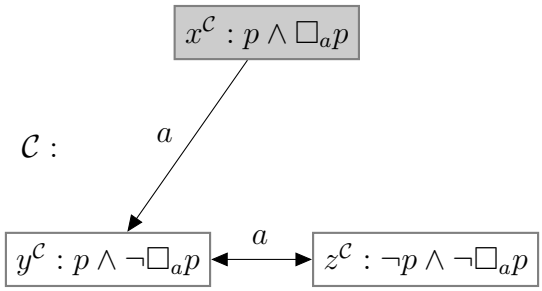
It is not even an atom over the set of formulas $\{\Box_a p\}$, because p is a subformula of $\Box_a p$.

On the other hand, in the following action model all preconditions are conjunctions of atoms over $\{p, \Box_a p\}$:



However, this model is still not canonical because there is an a-relation from $x^{\mathcal{B}}$ to $z^{\mathcal{B}}$, even though $p \wedge \Box_a p \wedge \Diamond_a(\neg p \wedge \neg\Box_a p)$ is inconsistent.

The following model does not have any of these inconsistent relations:



This model is canonical. All its preconditions are conjunctions of atoms over $\{p, \Box_a p\}$ and all its relations are consistent. Note that not all atoms are represented in the model: $\neg p \wedge \Box_a p$ is not present. Also, not all consistent relations are present: for example, there is no relation from $y^{\mathcal{C}}$ to $x^{\mathcal{C}}$, even though this would be allowed.

The nice thing about canonical action models is that each event completely determines the truth value of all formulas in Φ. In this section I will construct a notion of action emulation that corresponds to action model equivalence for canonical action models. But first I will show that every action model has an equivalent canonical action model.

7.3.12. Theorem. *Every finite action model has an equivalent canonical action model.*

PROOF. Take an action model $\mathcal{A} = (E, Pre, R, E_0)$. Let Σ be the set of atoms over $\Lambda_{\mathcal{A}}$. I construct a new action model $\mathcal{A}^c = (E^c, Pre^c, R^c, E_0^c)$ as follows:

$$\begin{array}{lll} E^c & := & \{(x,\sigma) \mid x \in E, \sigma \in \Sigma, Pre(x) \in \sigma\}, \\ Pre^c(x,\sigma) & := & \bigwedge \sigma, \\ (x,\sigma) \xrightarrow{a} (x',\sigma') & \text{iff} & x \xrightarrow{a} x' \text{ and } \bigwedge \sigma \wedge \Diamond_a \bigwedge \sigma' \text{ is consistent}, \\ E_0^c & := & \{(x,\sigma) \in E^c \mid x \in E_0\}. \end{array}$$

It follows from this definition that $\mathcal{A}^c$ is canonical. I claim that $\mathcal{A} \equiv \mathcal{A}^c$.

Take some model $\mathcal{M}$. Define a relation Z on $\mathcal{M} \otimes \mathcal{A} \times \mathcal{M} \otimes \mathcal{A}^c$ as follows:

$$(w,x)Z(v,(y,\sigma)) \text{ iff } w = v \text{ and } x = y.$$

I will start out by showing that Z is total. Take some $(w,x) \in W_{\mathcal{M}\otimes\mathcal{A}}$. Let $\sigma = \{\varphi \in \Lambda_{\mathcal{A}} \mid \mathcal{M} \models_w \varphi\}$. Then $\sigma \in \Sigma$ and $Pre(x) \in \sigma$ so $(x,\sigma) \in E^c$. Clearly, $\mathcal{M} \models_w \bigwedge \sigma$ so $(w,(x,\sigma)) \in W_{\mathcal{M}\otimes\mathcal{A}^c}$ and $(w,x)Z(w,(x,\sigma))$. Now take some $(w,(x,\sigma)) \in W_{\mathcal{M}\otimes\mathcal{A}^c}$. By definition of $\mathcal{A}^c$, $\mathcal{M} \models_w \bigwedge \sigma$ and $Pre(x) \in \sigma$ so $\mathcal{M} \models_w Pre(x)$ and $(w,x)Z(w,(x,\sigma))$.

Now I will show that Z is a bisimulation. Suppose $(w,x)Z(w,(x,\sigma))$. Invariance is satisfied because both (w,x) and $(w,(x,\sigma))$ inherit their valuation from w. For Zig, suppose $(w,x) \xrightarrow{a} (w',x')$. Let $\sigma' = \{\varphi \in \Lambda_{\mathcal{A}} \mid \mathcal{M} \models_{w'} \varphi\}$. By definition of Z, $\mathcal{M} \models_w \bigwedge \sigma$ and clearly $\mathcal{M} \models_{w'} \bigwedge \sigma'$ so $\bigwedge \sigma \wedge \Diamond_a \bigwedge \sigma'$ is consistent. Then by definition of R^c I have $(x,\sigma) \xrightarrow{a} (x',\sigma')$ so $(w,(x,\sigma)) \xrightarrow{a} (w',(x',\sigma'))$. Furthermore, $(w',x')Z(w',(x',\sigma'))$. This shows satisfaction of Zig.

For Zag, suppose $(w,(x,\sigma)) \xrightarrow{a} (w',(x',\sigma'))$. Then $w \xrightarrow{a} w'$ and $x \xrightarrow{a} x'$ so $(w,x) \xrightarrow{a} (w',x')$. Furthermore, $(w',x')Z(w',(x',\sigma'))$. This shows the satisfaction of Zag. □

So for every world in the original model, I construct the possible atoms corresponding to that world. I preserve only the relations from the original model that are consistent. This way I construct an equivalent canonical action model. Note that in the previous example, the action model $\mathcal{C}$ would be the result of constructing equivalent canonical models for $\mathcal{A}$ and $\mathcal{B}$ in this manner.

Now I will define a new notion of action emulation. I will use some notation adopted from [van Eijck et al., 2012]:

- If $\xrightarrow{a}$ is a relation on $X \times Z$, $x \in X$ and $Y \subseteq Z$ then I write $x \xrightarrow{\bar{a}} Y$ to mean that $x \xrightarrow{a} y$ for every $y \in Y$,
- If E is a relation on $X \times Z$, $x \in X$ and $Y \subseteq Z$ then I write $x \overrightarrow{E} Y$ to mean that xEy for every $y \in Y$,
- If E is a relation on $Z \times Y$, $X \subseteq Z$ and $y \in Y$ then I write $X \overrightarrow{E} y$ to mean that xEy for every $x \in X$.

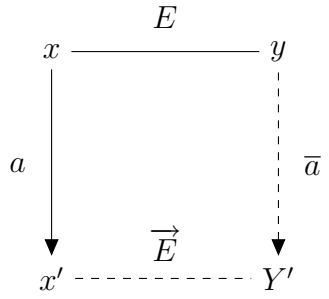
7.3.13. Definition. Given two finite action models $\mathcal{A}$ and $\mathcal{B}$, a relation $E : E^{\mathcal{A}} \times E^{\mathcal{B}}$ is an action emulation if for any $x \in E^{\mathcal{A}}, y \in E^{\mathcal{B}}$ such that xEy the following hold:

Consistency $Pre(x) \wedge Pre(y)$ is consistent.

Zig If $x \xrightarrow{a} x'$ then there is $Y' \subseteq E_{\mathcal{B}}$ such that $y \xrightarrow{\overline{a}} Y'$, $x' \overrightarrow{E} Y'$ and

$$Pre(x) \wedge Pre(y) \models \Box_a(Pre(x') \rightarrow \bigvee_{y' \in Y'} Pre(y')).$$

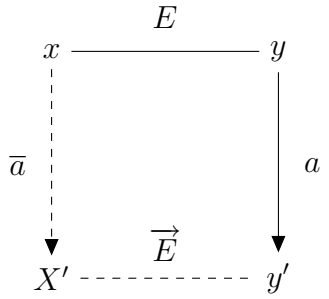
In a picture:



Zag If $y \xrightarrow{a} y'$ then there is $X' \subseteq E_{\mathcal{A}}$ such that $x \xrightarrow{\overline{a}} X'$, $X' \overrightarrow{E} y'$ and

$$Pre(x) \wedge Pre(y) \models \Box_a(Pre(y') \rightarrow \bigvee_{x' \in X'} Pre(x')).$$

In a picture:



I will say that $\mathcal{A}$ and $\mathcal{B}$ emulate, notation $\mathcal{A} \leftrightarrows \mathcal{B}$, if there is an action emulation E such that for every $x \in E_0^{\mathcal{A}}$ there is $Y \subseteq E_0^{\mathcal{B}}$ such that $x \overrightarrow{E} Y$ and $Pre(x) \models \bigvee_{y \in Y} Pre(y)$, and vice versa.

So if $\mathcal{A}$ and $\mathcal{B}$ emulate, every event in $\mathcal{A}$ corresponds to a number of events in $\mathcal{B}$, and vice versa. The preconditions of corresponding events are consistent with

each other. Furthermore, if x corresponds to y then any relation from x to a new event x' is matched by a relation from y to a set Y'. This set is chosen such that if a world of a Kripke model matches x and y and has a successor that matches x', then this successor also matches a member of Y'.

This notion of action emulation is sufficient for action model equivalence.

7.3.14. Theorem. *For any two finite action models $\mathcal{A}$ and $\mathcal{B}$, if $\mathcal{A} \leftrightarrows \mathcal{B}$ then $\mathcal{A} \equiv \mathcal{B}$.*

Proof. Suppose $\mathcal{A} \leftrightarrows \mathcal{B}$ and let E be an action emulation between $\mathcal{A}$ and $\mathcal{B}$. Let $\mathcal{M}$ be an arbitrary Kripke model. I define a relation Z on $\mathcal{M} \otimes \mathcal{A} \times \mathcal{M} \otimes \mathcal{B}$ as follows:

$$(w, x)Z(v, y) \text{ iff } w = v \text{ and } xEy.$$

I will first show that this relation is total on the actual worlds of $\mathcal{M} \otimes \mathcal{A}$ and $\mathcal{M} \otimes \mathcal{B}$. Recall that $U_{\mathcal{M}\otimes\mathcal{A}}$ is the set of actual worlds of the model $\mathcal{M} \otimes \mathcal{A}$. Suppose $(w, x) \in U_{\mathcal{M}\otimes\mathcal{A}}$. Then $x \in E_0^{\mathcal{A}}$ so there must be some $Y \subseteq E_0^{\mathcal{B}}$ such that $x \overrightarrow{E} Y$ and $Pre(x) \models \bigvee_{y\in Y} Pre(y)$. Then $\mathcal{M} \models_w \bigvee_{y\in Y} Pre(y)$, so there is some $y \in Y$ such that $\mathcal{M} \models_w Pre(y)$. But then $(w, x)Z(w, y)$. The proof for the other direction is analogous, so I conclude that Z is total.

Next, I will show that Z is a bisimulation. Suppose $(w, x)Z(w, y)$. Then xEy. Invariance is satisfied because both (w, x) and (w, y) inherit their valuation from w. For zig, suppose $(w, x) \xrightarrow{a} (w', x')$. Then $x \xrightarrow{a} x'$. By the fact that xEy there must be $Y' \subseteq E^{\mathcal{B}}$ such that $y \xrightarrow{\overline{a}} Y'$, $x \overrightarrow{E} Y'$ and

$$Pre(x) \wedge Pre(y) \models \Box_a(Pre(x') \rightarrow \bigvee_{y'\in Y'} Pre(y')).$$

It holds that $\mathcal{M} \models_w Pre(x) \wedge Pre(y)$ and $\mathcal{M} \models_{w'} Pre(x')$ and this gives $\mathcal{M} \models_{w'} \bigvee_{y'\in Y'} Pre(y')$, so there must be some $y' \in Y'$ such that $\mathcal{M} \models_{w'} Pre(y')$. Because $y' \in Y'$ it holds that $y \xrightarrow{a} y'$ and $x'Ey'$ so $(w, y) \xrightarrow{a} (w', y')$ and $(w, x')Z(w, y')$. This shows the satisfaction of Zig. The proof for Zag is analogous, so I conclude that $\mathcal{M} \otimes \mathcal{A} \leftrightarrows \mathcal{M} \otimes \mathcal{B}$ and, because $\mathcal{M}$ was arbitrary, $\mathcal{A} \equiv \mathcal{B}$. □

This result gives one half of a correspondence between action emulation and action model equivalence.

Turning to the other half, I will show that for canonical action models, action emulation is also necessary for action model equivalence.

7.3.15. Theorem. *If $\mathcal{A}$ and $\mathcal{B}$ are canonical and $\mathcal{A} \equiv \mathcal{B}$ then $\mathcal{A} \leftrightarrows \mathcal{B}$.*

PROOF. Suppose $\mathcal{A}$ and $\mathcal{B}$ are canonical and $\mathcal{A} \equiv \mathcal{B}$. Let $\mathcal{M}$ be the canonical Kripke model over $\Lambda_{\mathcal{A}} \cup \Lambda_{\mathcal{B}}$. Since $\mathcal{A} \equiv \mathcal{B}$, by Lemma 7.3.7 there is a bisimulation Z between $\mathcal{M} \otimes \mathcal{A}$ and $\mathcal{M} \otimes \mathcal{B}$ such that $(w,x)Z(v,y)$ implies $w = v$. Define a relation $E : E_{\mathcal{A}} \times E_{\mathcal{B}}$ as follows:

$$xEy \text{ iff } \exists w \in W_{\mathcal{M}} : (w,x)Z(w,y).$$

I will show that E is an action emulation. Suppose xEy and $(w,x)Z(w,y)$. I know that $Pre(x) \wedge Pre(y)$ is consistent because $\mathcal{M} \models_w Pre(x) \wedge Pre(y)$. Suppose $x \xrightarrow{a} x'$.

I need to show that there is a set Y' such that $y \xrightarrow{\bar{a}} Y'$, $x' \overrightarrow{E} Y'$ and $Pre(x) \wedge Pre(y) \models \Box_a(Pre(x') \rightarrow \bigvee_{y' \in Y'} Pre(y'))$. Let

$$Y' := \{y' \in E_{\mathcal{B}} \mid \exists w' \in W_{\mathcal{M}} : \begin{array}{l}(w,x) \xrightarrow{a} (w',x'),\\ (w,y) \xrightarrow{a} (w',y'),\\ (w',x')Z(w',y')\}.\end{array}$$

It follows from the definition of Y' that $y \xrightarrow{\bar{a}} Y'$ and $x' \overrightarrow{E} Y'$.

Now I need to show that $Pre(x) \wedge Pre(y) \models \Box_a(Pre(x') \rightarrow \bigvee_{y' \in Y'} Pre(y'))$. Suppose there is some model N and worlds $v, v' \in W_N$ such that $N \models_v Pre(x) \wedge Pre(y)$, $v \xrightarrow{a} v'$ and $N \models_{v'} Pre(x')$. Let $w' := \bigcup\{\varphi \in \Lambda_{\mathcal{A}} \cup \Lambda_{\mathcal{B}} \mid N \models_{v'} \varphi\}$. Then $w' \in W_{\mathcal{M}}$ and $Pre(x) \wedge Pre(y) \wedge \Diamond_a w'$ is consistent. Note that because $\mathcal{A}$ is canonical over $\Lambda_{\mathcal{A}}$, $\mathcal{B}$ over $\Lambda_{\mathcal{B}}$ and $\mathcal{M}$ over $\Lambda_{\mathcal{A}} \cup \Lambda_{\mathcal{B}}$, each world in $\mathcal{M}$ is completely determined by matching an event from $\mathcal{A}$ and one from $\mathcal{B}$. So since $\mathcal{M} \models_w Pre(x) \wedge Pre(y)$, $w \equiv Pre(x) \wedge Pre(y)$. So $w \wedge \Diamond_a w'$ is consistent, and because $\mathcal{M}$ is canonical, $w \xrightarrow{a} w'$. Since $Pre(x') \in w'$ then $(w,x) \xrightarrow{a} (w',x')$. Since $(w,x)Z(w,y)$ then there must be y' such that $(w,y) \xrightarrow{a} (w',y')$ and $(w',x')Z(w',y')$. Then $y' \in Y'$ and $Pre(y') \in w'$, so $N \models_{v'} Pre(y')$ and $N \models_{v'} \bigvee_{y' \in Y'} Pre(y')$. I conclude that $Pre(x) \wedge Pre(y) \models \Box_a(Pre(x') \rightarrow \bigvee_{y' \in Y'} Pre(y'))$. The proof for Zag is analogous. This shows that E is an action emulation.

To see that E is total on the actual events of $\mathcal{A}$ and $\mathcal{B}$, suppose $x \in E_0^{\mathcal{A}}$. Let $W_x = \{w \in W_{\mathcal{M}} \mid \mathcal{M} \models_w Pre(x)\}$. By totality of Z and the fact that $(w,x)Z(v,y)$ implies $w = v$ I have that for evey $w \in W$ there is an y such that $(w,x)Z(w,y)$. Let $Y = \{y \in E_{\mathcal{B}} \mid \exists w \in W_x : (w,x)Z(w,y)\}$. Then $x \overrightarrow{E} Y$ and

$$\begin{array}{ll} Pre(x) & \models \bigvee_{w \in W} w \text{ and} \\ \bigvee_{w \in W} w & \models \bigvee_{y \in Y} Pre(y), \text{ so} \\ Pre(x) & \models \bigvee_{y \in Y} Pre(y). \end{array}$$

The proof for totality in the other direction is analogous. This shows that $\mathcal{A} \leftrightarrows \mathcal{B}$. $\Box$

Together this gives:

7.3.16. Theorem. *For any two canonical action models $\mathcal{A}$ and $\mathcal{B}$,*

$$\mathcal{A} \equiv \mathcal{B} \textit{ iff } \mathcal{A} \leftrightarrows \mathcal{B}.$$

So for canonical action models, action emulation characterizes action model equivalence. This gives a procedure to check whether any two action models are equivalent: just compute the corresponding canonical action models and check whether there is an emulation between them. This is less work than computing the canonical Kripke model as is necessary for checking the existence of a parameterized action emulation, since not all atoms are represented in the canonical action model. Sometimes it may not even be necessary to compute the canonical action model: I have shown that action emulation is sufficient for action equivalence in the general case. So if there is already an action emulation between two non-canonical action models, there is no need to compute the corresponding canonical action models.

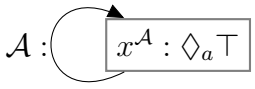
7.4 Propositional Action Emulation

In this section, I will compare my notion of action emulation to the notion of propositional action emulation presented in [van Eijck et al., 2012]. It is shown there that propositional action emulation corresponds to action model equivalence for a restricted class of action models, namely the propositional action models.

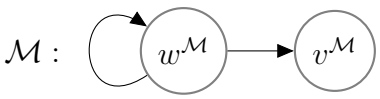
7.4.1. Definition. An action model is propositional if all preconditions of its events are formulas of classical propositional logic.

Unlike the class of canonical action models, this is a proper subclass of the class of all action models. It is not possible to find for every non-propositional action model an equivalent propositional one.

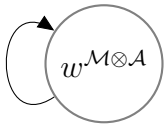
7.4.2. Example. Consider the following action model:



This action model selects all worlds that have an a-successor. There is no way to construct an equivalent action model that has only propositional preconditions. The following Kripke model demonstrates this:



The result of updating this model with $\mathcal{A}$ is the shown below.

$\mathcal{M} \otimes \mathcal{A}$: 

In the result, the world w is preserved because it has an a-successor. The world v is removed because it has no successors. There is no propositional difference between w and v, so any propositional action model that preserves w will also preserve v. Furthermore, the model $\mathcal{M} \otimes \mathcal{A}$ is not bisimilar to any result of an update in which v is preserved, because there are no relations departing from v so v is not bisimilar to $w^{\mathcal{M}\otimes\mathcal{A}}$. In other words, $\mathcal{M} \otimes \mathcal{A}$ is not bisimilar to the result of the update of $\mathcal{M}$ with a propositional action model. Therefore $\mathcal{A}$ is not equivalent to a propositional action model.

This example shows that the class of propositional action models is indeed a proper subclass of the class of all action models.

Now I will give the definition of propositional action emulation.

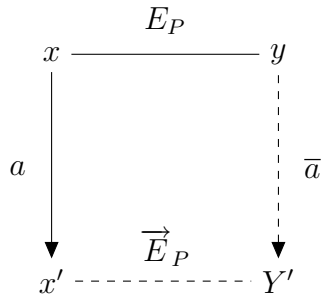
7.4.3. Definition. Given two finite action models $\mathcal{A}$ and $\mathcal{B}$, a relation $E_P : E^{\mathcal{A}} \times E^{\mathcal{B}}$ is an action emulation if for any $x \in E^{\mathcal{A}}, y \in E^{\mathcal{B}}$ such that xE_Py the following hold:

Consistency $Pre(x) \wedge Pre(y)$ is consistent.

Zig If $x \xrightarrow{a} x'$ then there is a non-empty set $Y' \subseteq E_{\mathcal{B}}$ such that $y \xrightarrow{\overline{a}} Y'$, $x' \overrightarrow{E}_P Y'$ and

$$Pre(x') \models \bigvee_{y' \in Y'} Pre(y').$$

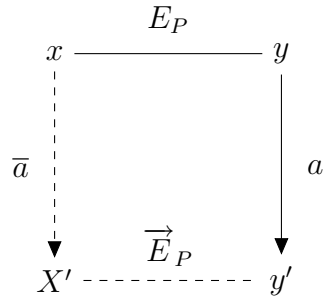
In a picture:



Zag If $y \xrightarrow{a} y'$ then there is a non-empty set $X' \subseteq E_{\mathcal{A}}$ such that $x \xrightarrow{\overline{a}} X'$, $X' \overrightarrow{E}_P y'$ and

$$Pre(y') \models \bigvee_{x' \in X'} Pre(x').$$

In a picture:



I will say that $\mathcal{A}$ and $\mathcal{B}$ propositionally emulate, notation $\mathcal{A} \leftrightarrows_P \mathcal{B}$, if for every $x \in E_0^{\mathcal{A}}$ there is $Y \subseteq E_0^{\mathcal{B}}$ such that $x \overrightarrow{E}_P Y$ and $Pre(x) \models \bigvee_{y \in Y} Pre(y)$, and vice versa.

It is shown in [van Eijck et al., 2012] that for propositional action models, propositional action emulation corresponds to action model equivalence.

7.4.4. Theorem. *For propositional action models $\mathcal{A}$ and $\mathcal{B}$,*

$$\mathcal{A} \equiv \mathcal{B} \text{ iff } \mathcal{A} \leftrightarrows_P \mathcal{B}.$$

I will now compare my notion of action emulation to the notion of propositional action emulation. The main difference is in the Zig and Zag conditions, more specifically in the constraint on the preconditions of the events in the sets X' and Y'. For propositional action emulation, the constraint for the Zig case is:

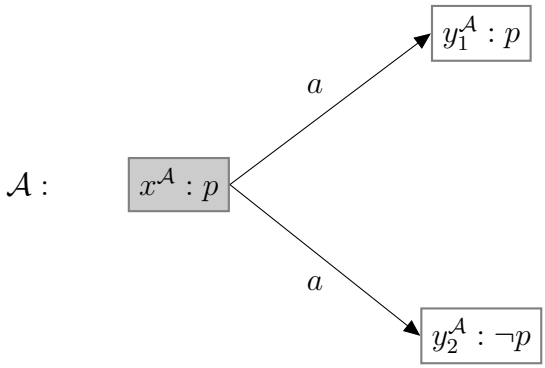
$$Pre(x') \models \bigvee_{y' \in Y'} Pre(y').$$

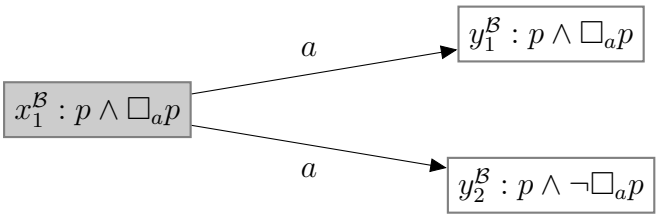
So every world that matches x' should also match one of the events in Y'. This condition assures that whenever a world is matched by a successor x' of x then it is also matched by a successor in Y' of y'. However, this condition also constrains worlds that match x' but are *not* a successor of a world that matches x. Therefore, I think this condition is too strong. In my definition of action emulation I use a weaker condition:

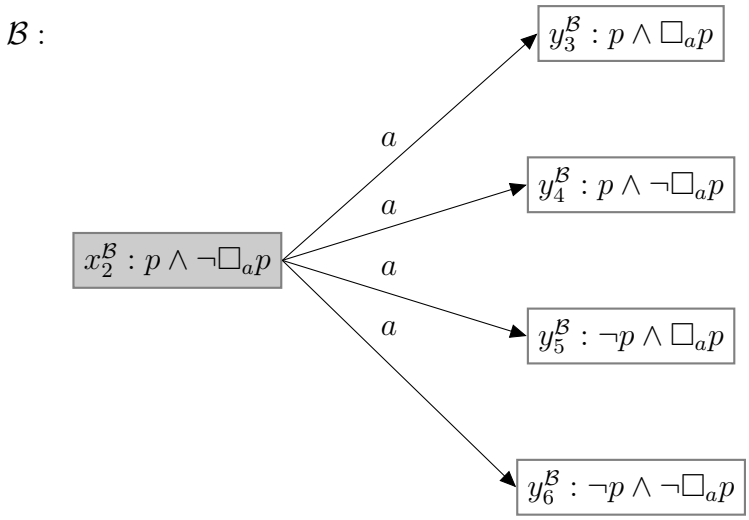
$$Pre(x) \wedge Pre(y) \models \Box_a(Pre(x') \rightarrow \bigvee_{y' \in Y'} Pre(y')).$$

This condition states that if a world matches x and y then all its successors that match x' match one of the worlds in Y'. This way it *only* constrains the worlds that are successors of worlds that match both x and y. This more subtle condition says exactly what is needed to define action equivalence between canonical models. The fact that the first condition is too strong is shown by the following example.

7.4.5. Example. Consider the following two action models:







These action models are canonical and equivalent, but they do not propositionally emulate.

To see that these models are equivalent, suppose that some world w matches the event $x^{\mathcal{A}}$ in the first model $\mathcal{A}$. If w satisfies $\Box_a p$ then it will match $x_1^{\mathcal{B}}$ in $\mathcal{B}$ and otherwise it will match $x_2^{\mathcal{B}}$ in $\mathcal{B}$. Suppose w has some successor that matches $y_1^{\mathcal{A}}$. Then this successor satisfies p so it will match either $y_1^{\mathcal{B}}$ or $y_2^{\mathcal{B}}$ if w matched

$x_1^{\mathcal{B}}$, or $y_3^{\mathcal{B}}$ or $y_4^{\mathcal{B}}$ if w matched $x_2^{\mathcal{B}}$. Suppose w has some successor that matches $y_2^{\mathcal{A}}$. Then this successor does not satisfy p, so w does not satisfy $\Box_a p$, so w matched $x_2^{\mathcal{B}}$. In this case the successor of w will match $y_5^{\mathcal{B}}$ or $y_6^{\mathcal{B}}$.

Another way to see that these canonical models are equivalent is by checking that the relation given by

$$E = \{(x^{\mathcal{A}}, x_1^{\mathcal{B}}), (y_1^{\mathcal{A}}, y_1^{\mathcal{B}}), (y_1^{\mathcal{A}}, y_2^{\mathcal{B}}), (x^{\mathcal{A}}, x_2^{\mathcal{B}}), (y_1^{\mathcal{A}}, y_3^{\mathcal{B}}), (y_1^{\mathcal{A}}, y_4^{\mathcal{B}}), (y_2^{\mathcal{A}}, y_5^{\mathcal{B}}), (y_2^{\mathcal{A}}, y_6^{\mathcal{B}})\}$$

is an action emulation between $\mathcal{A}$ and $\mathcal{B}$.

To see that the models do not propositionally emulate, observe that $x_1^{\mathcal{B}}$ does not emulate with $x^{\mathcal{A}}$ (or any other event in $\mathcal{A}$). This is because from $x^{\mathcal{A}}$ there is a relation to $y_2^{\mathcal{A}}$, while there is no set of successors of $x_1^{\mathcal{B}}$ such that the precondition $\neg p$ implies the disjunction of preconditions of events in this set.

This shows that propositional action emulation does not characterize action equivalence between canonical action models, nor action model equivalence between action models in general.

7.5 Conclusion

In this chapter I studied the properties of action models. Action models are applied on Kripke models and they are equivalent if they give equivalent results for all possible Kripke models. I tried to find a relation between action models that signifies when they are equivalent, just like bisimulation does for Kripke models.

Finding an appropriate relation that signifies equivalence of action models is complicated by the fact that multiple worlds in the Kripke model may match one world in the action model, and vice versa. I circumvent this complication by considering canonical action models. My main result is a notion of action emulation that is sufficient for action model equivalence of general action models. For canonical action models, this notion of action emulation is also necessary for equivalence. Because every action model has an equivalent canonical action model this gives a method to determine whether any two action models are equivalent. One can first try to find an action emulation between the models, which is already sufficient for equivalence. If that does not succeed one can construct the corresponding canonical action models and check whether there exists an action emulation between those, which gives a conclusive answer. The question of whether my notion of action emulation is equivalent to action model equivalence for all action models, not just the canonical ones, is left for future work.

I compared my notion of action emulation to two notions given in [van Eijck et al., 2012]: that of parameterized action emulation and that of propositional action emulation. My notion of action emulation has clear advantages compared

to both these notions. The advantage compared to parameterized action emulation is that there is no need to compute a separate relation for every world in the canonical Kripke model. This makes my method a lot more efficient. The advantage compared to propositional action emulation is that propositional action emulation only works for propositional action models, while my method works for all canonical action models. Because every action model has an equivalent canonical action model, this gives a solution for the entire class of action models.

Chapter 8

Knowledge, Belief and Preference

8.1 Introduction

Knowledge is often described by philosophers as justified true belief. In this chapter, I will investigate the interplay between knowledge and belief. I will propose a way to model different kinds of belief, one of which is knowledge, and show how this modeling procedure works out by analyzing a scenario of judgement aggregation in a Dutch meeting.

In [van Eijck and Wang, 2008] it is shown how propositional dynamic logic (PDL) can be interpreted as a logic of belief revision that extends the logic of communication and change (LCC) given in [van Benthem et al., 2006]. This new version of epistemic/doxastic PDL does not impose any constraints on the basic relations and because of this it does not suffer from the drawback of LCC that these constraints may get lost under updates that are admitted by the system.

Here, I will impose one constraint, namely that the agent's plausibility relations are linked. Linkedness is a natural extension of local connectedness to the multi-agent case and it ensures that the agent's preferences between all relevant alternatives are known. Since the belief updates that are used in [van Eijck and Wang, 2008] may not preserve linkedness, I will limit myself to a particular kind of belief change that does preserve it.

My framework has obvious connections to coalition logic [Pauly, 2002] and social choice theory [Taylor, 2005]. I will show how it can be used to model consensus seeking in plenary Dutch meetings. In Dutch meetings, a belief update is done for all agents in the meeting if a majority believes the proposition that is under discussion. A special case of these meetings is judgement aggregation, and I will apply my framework to the discursive dilemma in this field.

The discursive dilemma is considered in [List and Pettit, 2005]. This problem is the case of three judges a, b, c with a, b agreeing that p, and b, c agreeing that q, so that both p and q command a majority, but $p \wedge q$ does not. The example shows that majority judgement is not closed under logical consequence. To see

the relevance of the example for the practice of law, assume that p expresses that the defendant has done action X, and q expresses that the defendant is under a legal obligation not to do X. Then $p \wedge q$ expresses that the defendant has broken his contract not to do X. This is a standard paradox in judgement aggregation called the discursive dilemma or doctrinal paradox.

The discursive dilemma is an example of a situation where multiple agents have different beliefs. I will present an epistemic/doxastic framework that can be used to model such situations, and present a way to update these frameworks with new beliefs. In the above example, this gives a protocol for judgement aggregation.

In the previous chapters, I interpreted the relations of my models as knowledge relations for the agents. In this chapter, I allow for multiple interpretations. The relations can be seen as plausibility relations representing the belief of the agents which relates my approach to epistemic logic and the knowledge relations in the other chapters. They can also be seen as representing the preference of the agents, which connects my work to social choice theory. In the rest of this chapter I will refer to the relations as 'preference relations', but I do not wish to exclude other interpretations.

8.2 Belief Revision Without Constraints

In this section I will introduce a logic that is interpreted on Kripke models with preference relations. To start out with, there are no constraints on these preference relations. In particular, they do not need to be equivalence relations. This is exactly what makes the difference between knowledge and belief. I will also show how knowledge relations which are reflexive, symmetric and transitive can be constructed from these preference relations by using PDL.

Kripke models are defined in Chapter 2. In order to make a distinction between knowledge and preference relations, I will refer to the relations of a Kripke model $\mathcal{M}$ as $P^{\mathcal{M}}$ rather than $R^{\mathcal{M}}$.

When the relations of the Kripke models were interpreted as the agent's knowledge relations, a relation between two worlds meant that in one world, the agent considered the other one possible. Now, a relation from w to v means that in world w, the agent considers v possible and *at least as plausible* or *at least as preferred* as w.

The logic I will use is very much like the language presented in Chapter 2. Let $\mathcal{L}_{Pr}$ be the language with $\phi \in \mathcal{L}_{Pr}$ defined as follows:

$$\begin{array}{lcll} \phi & ::= & p \mid \neg\phi \mid \phi \vee \phi \mid \langle\alpha\rangle\phi & \text{where } p \in P, \\ \alpha & ::= & a \mid a^{\smile} \mid ?\phi \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha^* & \text{where } a \in Ag. \end{array}$$

This language is interpreted as defined in Chapter 2. The only new construct is the program $a^{\smile}$. This expresses the ***converse*** of a: if there is an a-relation

from w to v then there is an $a^{\smile}$ relation from v to w. In the case of knowledge relations, this construct would be quite useless because all relations would be symmetric. Now, it is a very useful construct that expresses that the $a^{\smile}$-related world is considered at most at preferable or at most as plausible as the current world by agent a. Recall that given some program α, $[\![\alpha]\!]^{\mathcal{M}}$ denotes the relation that interprets the program α in $\mathcal{M}$.

As mentioned above, if there is a relation from v to w then the agent considers v at least as plausible or preferable as w. If there is also a direct or indirect path from v back to w, then the agent considers both worlds equally plausible or preferable. If there is no path from v to w, then the agent considers v more plausible or preferable than w.

Using the programs α, one can express a great number of different notions of belief and knowledge. I will focus on knowledge, strong belief, plain belief and conditional belief.

Knowledge an agent knows something if it holds in all possible worlds, regardless of how plausible or preferable these worlds are. I construct an equivalence relation representing knowledge by constructing the union of the preference relation with its converse, and taking the reflexive transitive closure of the result. This gives:

$$\sim_a := (a \cup a^{\smile})^*.$$

The formula $[\sim_a]\phi$ expresses that agent a knows ϕ.

Strong belief an agent strongly believes something if it holds in all worlds that he considers at least as plausible or preferable as the current world. The relation for strong belief is constructed by taking the reflexive transitive closure of the preference relation:

$$\geq_a := a^*.$$

The formula $[\geq_a]\phi$ expresses that a strongly believes that ϕ.
Note that since the relations point to the more preferred worlds, $w \geq_a v$ means that v is *at least as* preferred as w.

Plain belief an agent has plain belief in ϕ if it holds in the worlds the agents considers most plausible or preferable. This holds if there is some world the agent considers possible, such that all worlds at least as plausible as that world satisfy ϕ. One could think of that world as the least plausible world where ϕ holds. Therefore plain belief can be expressed as follows:

$$[\rightarrow_a]\phi \Leftrightarrow \langle\sim_a\rangle[\geq_a]\phi.$$

The formula $[\rightarrow_a]\phi$ expresses that a has plain belief in ϕ.

Conditional belief an agent believes ϕ conditional to ψ if he has plain belief that ϕ is true, given the fact that ψ holds. This holds if there is some ψ-world the agent considers possible, such that all ψ-worlds at least as plausible as that world satisfy ϕ. Trivially, it also holds if ψ is false. Conditional belief can be expressed as follows:

$$[\rightarrow_a^{\psi}]\phi \Leftrightarrow \langle\sim_a\rangle\psi \rightarrow \langle\sim_a\rangle(\psi \wedge [\geq_a](\psi \rightarrow \phi)).$$

The formula $[\rightarrow_a^{\psi}]\phi$ expresses that a has plain belief in ϕ, conditional to ψ. Note that plain belief can also be expressed as belief conditional to truth:

$$[\rightarrow_a]\phi \Leftrightarrow [\rightarrow_a^{\top}]\phi.$$

Any preference relation P_a can be turned into a pre-order by taking its reflexive transitive closure ${P_a}^*$. The abbreviation for strong belief introduces $\geq_a$ as names for these pre-orders. The knowledge abbreviation introduces $\sim_a$ as names for the equivalence relations given by $(P_a \cup {P_a}^{\smile})^*$.

The definition of $\rightarrow_a^{\phi}$ (conditional belief for a, with condition ϕ) is from [Boutilier, 1992]. This definition, also used in [Baltag and Smets, 2008], states that conditional to ϕ, a believes in ψ if either there are no accessible ϕ worlds, or there is an accessible ϕ world in which there is strong belief in $\phi \rightarrow \psi$. The definition of $\rightarrow_a^{\phi}$ matches the well-known accessibility relations $\rightarrow_a^V$ for each definable subset V of the domain, given by:

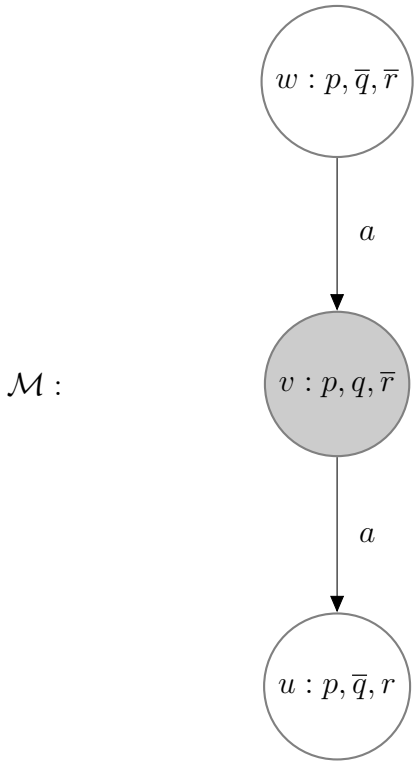
$$\rightarrow_a^V := \{(x, y) \mid x \sim_a y \wedge y \in \mathrm{MIN}_{\geq_a} V\},$$

where $\mathrm{MIN}_{\geq_a} V$, the set of minimal elements of V under $\geq_a$, is defined as

$$\{w \in V : \forall v \in v(v \geq_a w \Rightarrow w \geq_a v)\}.$$

Note that since $w \geq_a v$ expresses that v is at least as preferred as w, the elements of $MIN_{\geq_a}$ are the *most* preferred worlds, according to agent a.

8.2.1. Example. Consider the following model:



Here the relations represent the belief of agent a. In world v, agent a knows that p is true because it holds in all worlds she considers possible:

$$\mathcal{M} \models_v [\sim_a]p.$$

She has strong belief in $q \vee r$, since it holds in all worlds that she considers at least as plausible as v:

$$\mathcal{M} \models_v [\geq_a](q \vee r).$$

She has plain belief in r, since it holds in the world she considers most plausible:

$$\mathcal{M} \models_v [\rightarrow_a]r.$$

Finally, agent a believes that q holds given the fact that r does not hold, so she has plain belief in q conditional to $\neg r$. This holds because q is true in all preferred $\neg r$-worlds.

$$\mathcal{M} \models_v [\rightarrow_a^{\neg r}]q.$$

This logic is completely axiomatized by the standard PDL rules and axioms ([Segerberg, 1982, Kozen and Parikh, 1981]) plus the following axioms that describe the relation between the basic programs a and $a^{\smile}$:

$$\vdash \phi \rightarrow [a]\langle a^{\smile}\rangle\phi,$$
$$\vdash \phi \rightarrow [a^{\smile}]\langle a\rangle\phi.$$

If the P_a are well-founded, $\text{MIN}_{\leq_a} P$ will be non-empty for non-empty P. The canonical model construction for PDL yields finite models; since each relation on a finite model is well-founded, there is no need to impose well-foundedness as a relational condition.

This yields a very expressive complete and decidable PDL logic for belief revision, to which one can add mechanisms for belief update and for belief change.

Note that the definitions for knowledge and strong belief are given as single unary modalities $(a \cup a^{\smile})^*$ and a^*, while plain and conditional belief are defined in terms of the box modality. This is because in order to express plain and conditional belief as single unary modalities, I would have to extend the language of PDL with a new construct.

Suppose I would add the construct $\overline{\alpha}$ as a program, with the semantics that $w\overline{\alpha}v$ holds iff $w\alpha v$ does not hold. Let $\alpha - \beta$ be shorthand for the "subtraction" of β from α:

$$\alpha - \beta := \overline{\overline{\alpha} \cup \beta},$$

which holds between w and v iff α holds between w and v and β does not. Then I could express plain and conditional belief as single unary modalities as follows:

Plain belief plain belief could be expressed as a relation pointing to all the most preferred worlds. These are the worlds in which there is no strictly better world, according to $\geq_a$. In other words, there is no world reachable by a $\geq_a$-step that is not reachable by a $(\geq_a)^{\smile}$ step.

$$\rightarrow_a \quad := \quad \sim_a; ?([\geq_a - (\geq_a)^{\smile}]\bot).$$

Conditional belief belief conditional to ψ could be expressed as a relation pointing to all the most preferred ψ-worlds. These are the worlds in which there is no strictly better ψ-world, according to $\geq_a$. In other words, there is no ψ-world reachable by a $\geq_a$-step that is not reachable by a $(\geq_a)^{\smile}$ step.

$$\rightarrow_a^{\psi} \quad := \quad \sim_a; ?(\psi \wedge [\geq_a - (\geq_a)^{\smile}]\neg\psi).$$

Unfortunately, the logic of PDL with the complement operator is undecidable [Harel, 1984]. Therefore, I will not add the complement operator to my logic $\mathcal{L}_{Pr}$. Instead I will only use the $\rightarrow_a$ and $\rightarrow_a^{\psi}$ operators inside a box modality.

8.3 Belief Revision with Linked Preference Relations

The preference relations that serve as the basis for construction of a preference pre-order in Section 8.2 leave something to be desired. Compare an optometrist who collects answers for a number of lenses she tries out on you: "Better or worse?", (change of lens), "Better or worse?" (change of lens), "Better or

worse?"... . If you reply "worse" after a change of x to y, and "worse" after a change from y to z, she will most probably not bother to collect your reaction to a change from x to z. But what if you answer "better" after the second swap? Then, if she is reasonable, she will try to find out how x compares to z. It makes sense to impose this as a requirement on preference relations.

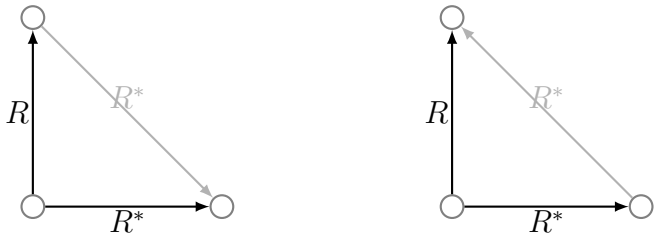
There are several ways to do this. Recall that I did not impose a requirement of transitivity on the basic preference relations. Here is a definition that does not imply transitivity, but yields that the transitive closures of the basic preference relations are well-behaved.

8.3.1. Definition. A binary relation R is **forward linked** if the following holds:

$$\forall x, y, z((xRy \wedge xR^*z) \rightarrow (yR^*z \vee zR^*y)).$$

R is **linked** if both R and $R^\smile$ are forward linked.

The following picture shows the idea, where one of the gray relations should be present whenever the black relations are:



Note that this is different from the notion of **weak connectedness**: a relation R is weakly connected if

$$\forall x, y, z((xRy \wedge xRz) \rightarrow (yRz \vee y = z \vee zRy)).$$

The following theorem shows the interplay between forward linkedness and weak connectedness.

8.3.2. Theorem. *R is forward linked iff R^* is weakly connected.*

Proof. The right to left direction is immediate. For the left to right direction, assume R is forward linked. Let wR^*w_1 and wR^*w_2. Then there is an $n \in \mathbb{N}$ with wR^nw_1. I will prove the claim by induction on n. If $n = 0$ then $w = w_1$ and $w_1R^*w_2$, and I am done. Otherwise, assume the claim holds for n. I have to show it holds for $n + 1$. Suppose $wR^{n+1}w_1$. Then for some w', $wRw'R^nw_1$. By forward linking of R, either $w'R^*w_2$ or w_2R^*w'. In the first case, use the induction hypothesis to get $w_1R^*w_2$ or $w_2R^*w_1$. In the second case, it follows from w_2R^*w' and $w'R^nw_1$ that $w_2R^*w_1$. □

Starting from relations that are linked, one can upgrade the method from the previous section to construct 'belief revision models' in the style of [Grove, 1988, Board, 2002, Baltag and Smets, 2006, 2008].

It is well-known that the following principle characterizes weak connectedness of P_a (cf. [Goldblatt, 1992]):

$$[a]((\phi \wedge [a]\phi) \rightarrow \psi) \vee [a]((\psi \wedge [a]\psi) \rightarrow \phi).$$

The notion of forward linking is characterized by:

$$[a]((\phi \wedge [a^*]\phi) \rightarrow \psi) \vee [a^*]((\psi \wedge [a^*]\psi) \rightarrow \phi). \qquad (*)$$

8.3.3. Theorem. *Principle (*) holds in a belief revision frame iff P_a is forward linked.*

Proof. Let (W, P) be a frame where P_a is forward linked, and let $\mathcal{M} = (W, P, V)$ be some model based on the frame. I will show that (*) holds. Let w be a world in $\mathcal{M}$. Assume $\mathcal{M} \not\models_w [a]((\phi \wedge [a^*]\phi) \rightarrow \psi)$. I have to show that $\mathcal{M} \models_w [a^*]((\psi \wedge [a^*]\psi) \rightarrow \phi)$. From the fact that $\mathcal{M} \not\models_w [a]((\phi \wedge [a^*]\phi) \rightarrow \psi)$, I get that there is a world w_1 with wP_aw_1 and $\mathcal{M} \models_{w_1} \phi \wedge [a^*]\phi \wedge \neg\psi$.

Let w_2 be an arbitrary world with $wP_a^*w_2$. Then by forward linking of P_a, either $w_1P_a^*w_2$ or $w_2P_a^*w_1$.

In the first case, it follows from $\mathcal{M} \models_{w_1} [a^*]\phi$ that $\mathcal{M} \models_{w_2} \phi$, and therefore $\mathcal{M} \models_{w_2} (\psi \wedge [a^*]\psi) \rightarrow \phi$. In the second case, it follows from $\mathcal{M} \models_{w_1} \neg\psi$ that $\mathcal{M} \models_{w_2} \neg[a^*]\psi$, and therefore $\mathcal{M} \models_{w_2} (\psi \wedge [a^*]\psi) \rightarrow \phi$. So in both cases, $\mathcal{M} \models_{w_2} (\psi \wedge [a^*]\psi) \rightarrow \phi$, and since w_2 was an arbitrary world with $wP_a^*w_2$, it follows that $\mathcal{M} \models_w [a^*]((\psi \wedge [a^*]\psi) \rightarrow \phi)$.

Next, assume a frame (W, P) where P_a is not forward linked. I will construct a model $\mathcal{M} = (W, P, V)$ and an instance of (*) that does not hold. If P_a is not forward linked, there are w, w_1, w_2 with wP_aw_1, $wP_a^*w_2$, and neither $w_1P_a^*w_2$ nor $w_2P_a^*w_1$. Construct the valuation of $\mathcal{M}$ by setting p true in w_1 and in all worlds w' with $w_1P_a^*w'$ and false everywhere else, and setting q true in w_2 and in all worlds w'' with $w_2P_a^*w''$, and false everywhere else. Note that since not $w_1P_a^*w_2$, p will be false in w_2, and that since not $w_2P_a^*w_1$, q will be false in w_1. So I get $\mathcal{M} \models_{w_1} p \wedge [a^*]p \wedge \neg q$ and $\mathcal{M} \models_{w_2} q \wedge [a^*]q \wedge \neg p$. It follows that

$$\mathcal{M} \models_w \langle a\rangle(p \wedge [a^*]p \wedge \neg q) \wedge \langle a^*\rangle(q \wedge [a^*]q \wedge \neg p),$$

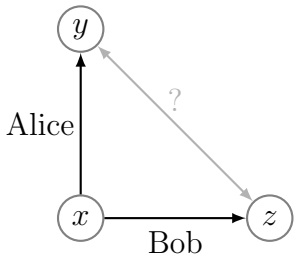
i.e.,

$$\mathcal{M} \not\models_w [a]((p \wedge [a^*]p) \rightarrow q) \vee [a^*]((q \wedge [a^*]q) \rightarrow p),$$

showing that this instance of (*) does not hold in $\mathcal{M}$. □

In the multi-agent case there is a further natural constraint. Consider a situation where Alice and Bob have to decide on the chairperson of a program

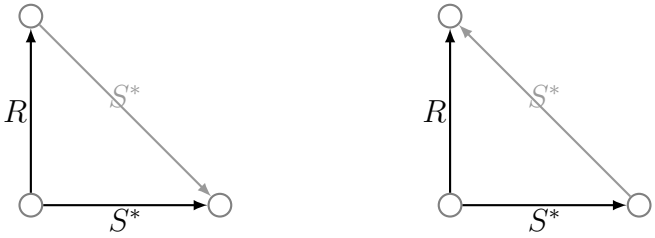
committee. Carol is mediator. Alice says she prefers y to x. Bob counters by saying that he prefers z to x. What should Carol do? Clearly, she should urge *both* of them to compare y and z.



Translating this example to my logic of belief, I want to require that if $x \geq_a y$ and $x \geq_b z$, then either $y \geq_a z$ or $z \geq_a y$ and either $y \geq_b z$ or $z \geq_b y$. This motivates the following extension of the definition of linkedness to the multi-agent case.

8.3.4. Definition. A set of binary relations $\mathbf{R}$ on a domain W is **forward linked** if for all R, S in $\mathbf{R}$, if xRy and xS^*z, then either yS^*z or zS^*y. $\mathbf{R}$ is **backward linked** if the set $\{R^\smile \mid R \in \mathbf{R}\}$ is forward linked. $\mathbf{R}$ is **linked** if $\mathbf{R}$ is both forward and backward linked.

The following picture shows the idea.



It follows from Definition 8.3.4 that the set $\{R\}$ is forward linked iff R is forward linked according to Definition 8.3.1. So Definition 8.3.4 gives a natural extension of linking (and of local connectedness) to the multi-agent case.

The following theorem shows that my definition satisfies the motivating requirement that if $x \geq_a y$ and $x \geq_b z$ then either $y \geq_a z$ or $z \geq_a y$:

8.3.5. Theorem. *If R and S are linked then for any x, y, z, if xR^*y and xS^*z then either yR^*z or zR^*y.*

PROOF. Suppose xR^*y and xS^*z. I will prove that for any w on the path from x to z, either wR^*y or yR^*w. This clearly holds for $w = x$. Suppose w is the successor of w' on the path, and the result holds for w'. Suppose $w'R^*y$. Since $w'Sw$ the result holds by forward linking of R and S. Suppose yR^*w'. $w'Sw$ and $w'R^*w'$ so either $w'R^*w$ or wR^*w'. In the first case trivially yR^*w. In the second case the result holds by backward linking of R. □

If one assumes that relations are linked, there is an interesting interplay between common knowledge and common belief. The following theorem shows that in this case common knowledge equals the union of strong common belief and strong reverse common belief:

8.3.6. THEOREM. *If R and S are linked, then*

$$(R \cup R^{\smile} \cup S \cup S^{\smile})^* = (R \cup S)^* \cup (R^{\smile} \cup S^{\smile})^*.$$

PROOF. The inclusion from right to left is obvious. For the inclusion from left to right, assume $x(R \cup R^{\smile} \cup S \cup S^{\smile})^*y$. Letting X and Y range over R and S, observe that each $X \circ Y^{\smile *}$ link can be replaced by either a Y^* or a $Y^{\smile *}$ link, and similarly for $X^{\smile} \circ Y^*$ links, by linking of R and S. Continuing this process until all one-step links are of the form $R \cup S$ or of the form $R^{\smile} \cup S^{\smile}$, this yields $x(R \cup S)^*y$ or $x(R^{\smile} \cup S^{\smile})^*y$. □

This theorem shows that linking of relations simplifies the notion of common knowledge.

The modal characterization of relation linking is given by:

$$[a]((\phi \wedge [b^*]\phi) \rightarrow \psi) \vee [b^*]((\psi \wedge [b^*]\psi) \rightarrow \phi) \qquad \text{(LINK)}$$

8.3.7. THEOREM. *The set of LINK principles (with a, b ranging over the set of all agents) holds in a belief revision model iff the basic plausibility relations in the model are forward linked.*

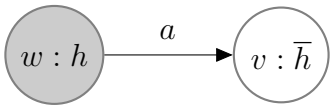
PROOF. Analogous to the proof of Theorem 8.3.3. □

8.4 Belief Update and Belief Change

In Chapter 2 I introduced action models and defined the update product. In [van Benthem et al., 2006] it is shown how extending the PDL language with a extra modality $[\mathcal{A}, e]\phi$ does not change its expressive power. The interpretation of the new modality is as follows: $[\mathcal{A}, e]\phi$ is true in w in $\mathcal{M}$ if success of the update of $\mathcal{M}$ with action model $\mathcal{A}$ to $\mathcal{M} \otimes \mathcal{A}$ implies that ϕ is true in (w, e) in

$\mathcal{M} \otimes \mathcal{A}$. The language of PDL with this new action update modality was called the Logic of Communication and Change or LCC. But LCC as it was proposed in [van Benthem et al., 2006] has a design flaw. It starts out with relations for the agents that are constrained in some way that is appropriate for notions of knowledge or belief. For example, KD45 models are often used to give a realistic reprsentation of belief. However, there is a problem with updating KD45 models. When a KD45 Kripke model is updated with a KD45 action model, the result may be a non-KD45 model. This means that the resulting relations cannot be interpreted as belief relations anymore. This issue is remedied in [van Eijck and Wang, 2008], where it was first proposed to construct the relational properties for belief from more basic relations by means of PDL operations. Here, I propose the same for the different notions of belief. Action update by means of the update construction can now be seen as belief update.

8.4.1. Example. Consider the following model of a situation where a coin has been tossed and agent a does not know the value of the coin. The proposition h signifies that the coin lies heads up, and agent a considers this less plausible than the situation where the coin lies tails up.



So in this example, agent a believes that the coin lies tails up. Now, if the model is updated with an action model that signifies that the coin lies heads up, the result is that world v disappears.

Belief change is something different from belief update. Belief update can only remove worlds and arrows. It can never reverse the direction of arrows or introduce new arrows, for the arrows in the update result are the arrows that are both in the original model and in the action model. Belief change is something more radical than this: replacing existing preference relations by new ones. Here, I will focus on belief change rather than belief update. Belief change can be compared to factual change. Factual change is what happens when the value of a proposition changes. For example, suppose a coin lies heads up which is signified by the truth of some proposition h. Now it is tossed again and it lies tails up. This is the factual change of $h = \top$ to $h = \bot$.

In [van Benthem et al., 2006], it was proposed to handle factual change by propositional substitution. I already used this in Chapter 3 to model the factual change that occurs when a message is sent in some message exchange. The factual change of the coin from heads to tails can be modeled as the propositonal substitution $\{h \mapsto \neg h\}$. Something similar can be done for belief change. Suppose agent a prefers x to y, she changes her preference, and now she prefers y to

x. Or suppose she reverses *all* her preferences. This can also be handled as a substitution, namely $\{a \mapsto a^{\smile}\}$.

Relational substitutions were proposed for belief change in [van Benthem, 2007], and it was shown in [van Eijck, 2008] that adding relational substitutions for preference change to epistemic PDL makes no difference for expressive power: the resulting system still reduces to PDL.

A preference substitution (or plausibility substitution) is a map from agents to programs that can be represented by a finite set of bindings

$$\{a_1 \mapsto \alpha_1, \ldots, a_n \mapsto \alpha_n\}$$

where the a_j are agents, all different, and where the α_i are programs. It is assumed that each a that does not occur in the left hand side of a binding is mapped to itself. Call the set $\{a \in Ag \mid \rho(a) \neq a\}$ the *domain* of ρ. If $\mathcal{M} = (W, P, V, W_0)$ is a preference model and ρ is a preference substitution, then $\mathcal{M}^\rho$ is the result of changing the preference map P of $\mathcal{M}$ to P^ρ given by:

$$P^\rho(a) \;:=\; \begin{cases} P_a \text{ for } a \text{ not in the domain of } \rho, \\ [\![\rho(a)]\!]^{\mathcal{M}} \text{ for } a \text{ in the domain of } \rho. \end{cases}$$

Now I will extend my PDL language with a modality $[\![\rho]\!]\phi$ for preference change, with the following interpretation:

$$\mathcal{M} \models_w [\![\rho]\!]\phi \text{ iff } \mathcal{M}^\rho \models_w \phi.$$

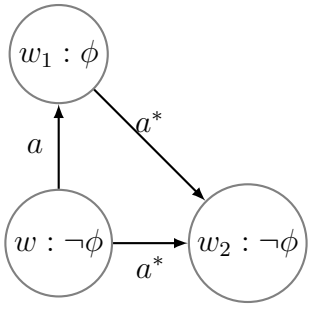
An important thing to note is that since there are constraints on the preference relations P_a (namely that they are linked), I need to ensure that the belief changing substitutions satisfy these constraints. Therefore, I will use the general definition of preference substitution to define an update that preserves linkedness.

Consider the suggestive upgrade $\sharp_a\phi$ discussed in [van Benthem and Liu, 2004]:

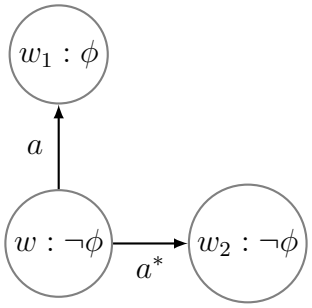
$$\sharp_a\phi := \; ?\phi; a; ?\phi \;\cup ?\neg\phi; a; ?\neg\phi \;\cup ?\neg\phi; a; ?\phi.$$

This is a variation on what is called the lexicographic upgrade in the belief revision community (see e.g., [Nayak, 1994]). The suggestive upgrade removes all relations from ϕ-worlds to $\neg\phi$-worlds. Belief revision with suggestive upgrade does not preserve linking of relations, as the following example shows.

8.4.2. Example. Consider a case where wP_aw_1 and $wP_a^*w_2$ and $w_1P_a^*w_2$, with ϕ true in w_1 but not in w and w_2.



This model is linked. After the suggestive upgrade for ϕ the a-path from w_1 to w_2 will be removed:



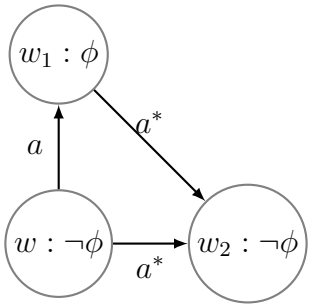
Clearly, now the model is not linked anymore.

So the suggestive upgrade does not preserve linking. However, if I revise the upgrade procedure so that it adds extra links instead of removing them, as follows, I get a variation that preserves linking:

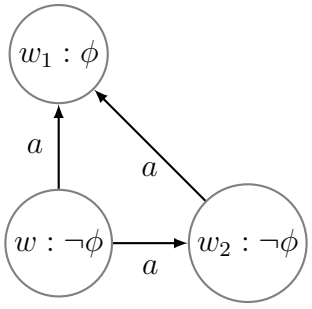
$$\natural_a\phi := \ ?\phi; a^*; ?\phi \ \cup ?\neg\phi; a^*; ?\neg\phi \ \cup ?\neg\phi; (a^* \cup a^{\breve{}*}); ?\phi.$$

Thus, instead of removing the relations from ϕ-worlds x to $\neg\phi$-worlds y, they get reversed, and extra links to x get added to 'support' the new link from y to x. Moreover, ϕ to ϕ links and $\neg\phi$ to $\neg\phi$ links are strengthened to deal with the problem of detours through worlds that assign a different truth value to ϕ.

8.4.3. Example. Consider again the linked model from the previous example.



If I apply the update $\natural_a\phi$ instead of $\sharp_a\phi$, I get the following result:



Now instead of removing the relation from w_1 to w_2 it has been reversed. Clearly this model is still linked, and belief in ϕ has been created.

The following theorem shows that the update $\natural_a\phi$ does preserve linkedness.

8.4.4. THEOREM. *If $\mathcal{M} = (W, P, V, W_0)$ is a belief revision model where P_a and P_b are linked, and ϕ is a PDL formula, then $[\![\natural_a\phi]\!]^{\mathcal{M}}$ and P_b are also linked.*

PROOF. Write a for P_a, b for P_b, and $\natural_a\phi$ for $[\![\natural_a\phi]\!]^{\mathcal{M}}$. First note that for any worlds x and y, if xa^*y then either $x(\natural_a\phi)y$ or $y(\natural_a\phi)x$.

Suppose xby and $x(\natural_a\phi)^*z$. I will show that either wa^*y or ya^*w for all w on the path from x to z.

Firstly let $w = x$. Since xby and xa^*x, either xa^*y or ya^*x by linking of a and b. Now let w' be the predecessor of w on the path, so $x(\natural_a\phi)^*w'$ and $w'(\natural_a\phi)w$. Suppose either ya^*w' or $w'a^*y$. Since $w'(\natural_a\phi)w$, either $w'a^*w$ or wa^*w'. If ya^*w' and $w'a^*w$ or wa^*w' and $w'a^*y$, then trivially ya^*w or wa^*y. Suppose $w'a^*y$ and $w'a^*w$. By forward linking of a and Theorem 8.3.2, wa^*y or ya^*w. Suppose ya^*w' and wa^*w'. By backward linking of a and Theorem 8.3.2, ya^*w or wa^*y. So then for any w on the path wa^*y or ya^*w, so za^*y or ya^*z, so $z(\natural_a\phi)y$ or $y(\natural_a\phi)z$.

Suppose $x(\natural_a\phi)y$ and xb^*z. Then either xa^*y or ya^*x. In the first case the result follows by Theorem 8.3.5. Suppose ya^*x. I will show that for any w on the path from x to z, yb^*w or wb^*y. Firstly let $w = x$. ya^*x and yb^*y so by Theorem 8.3.5 the result holds. Suppose w' is the predecessor of w on the path and the result holds for w'. Suppose yb^*w'. Then since $w'bw$, trivially yb^*w. Suppose $w'b^*y$. Then the result holds by linkedness of b. □

Now call a substitution where all bindings are of the form $a \mapsto \natural_a\phi$ a linked substitution. Then I construct a complete logic for belief change with linked substitutions, by means of reduction axioms that 'compile out' the belief changes (see [van Eijck, 2008], cf. Chapter 3):

8.4.5. THEOREM. *The logic of epistemic preference PDL with belief change modalities for linked substitutions is complete.*

PROOF. The preference change effects of $[\![\rho]\!]$ can be captured by a set of reduction axioms for $[\![\rho]\!]$ that commute with all sentential language constructs, and that handle formulas of the form $[\![\rho]\!][\pi]\phi$ by means of reduction axioms of the form

$$[\![\rho]\!][\pi]\phi \quad \leftrightarrow \quad [F_\rho(\pi)][\![\rho]\!]\phi,$$

with F_ρ given by:

$$\begin{array}{lcl} F_\rho(a) & := & \begin{cases} \rho(a) & \text{if } a \text{ in the domain of } \rho, \\ a & \text{otherwise,} \end{cases} \\ F_\rho(?\phi) & := & ?[\![\rho]\!]\phi, \\ F_\rho(\pi_1;\pi_2) & := & F_\rho(\pi_1);F_\rho(\pi_2), \\ F_\rho(\pi_1 \cup \pi_2) & := & F_\rho(\pi_1) \cup F_\rho(\pi_2), \\ F_\rho(\pi^*) & := & (F_\rho(\pi))^*. \end{array}$$

It is easy to check that these reduction axioms are sound, and that for each formula of the extended language the axioms yield an equivalent formula in which $[\![\rho]\!]$ occurs with lower complexity, which means that the reduction axioms can be used to translate formulas of the extended language to PDL formulas. Completeness then follows from the completeness of PDL. □

8.5 Analyzing Plenary Dutch Meetings

A plenary Dutch meeting (Dutch: 'Vergadering') is a simultaneous preference or belief change event where the following happens. Assume an epistemic situation $\mathcal{M}$ with actual world w, and assume proposition ϕ is on the agenda.

- If a majority prefers ϕ to $\neg\phi$, i.e., if

  $$|\{i \in Ag \mid \mathcal{M} \models_w [\rightarrow_i]\phi\}| > |\{i \in Ag \mid \mathcal{M} \models_w [\rightarrow_i]\neg\phi\}|$$

  then simultaneous belief or preference change $\{i \mapsto \natural_i\phi \mid i \in Ag\}$ takes place.

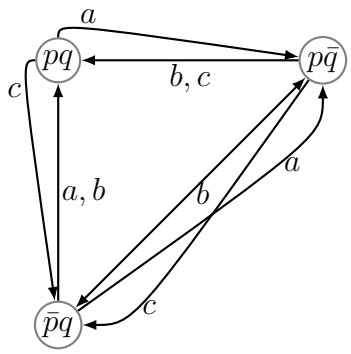
- If a majority prefers $\neg\phi$ to ϕ, i.e., if

  $$|\{i \in Ag \mid \mathcal{M} \models_w [\rightarrow_i]\phi\}| < |\{i \in Ag \mid \mathcal{M} \models_w [\rightarrow_i]\neg\phi\}|$$

  then simultaneous belief or preference change $\{i \mapsto \natural_i\neg\phi \mid i \in Ag\}$ takes place.

- If there is no majority either way, nothing happens.

In fact, Dutch meetings are procedures for judgement aggregation [List and Pettit, 2005]. Let me return to the example of three judges a, b, c with a, b agreeing that p, and b, c agreeing that q, so that both p and q command a majority, but $p \wedge q$ does not. Using my logic, I can picture the situation as a preference model. I assume that every agent has greater belief in worlds that match her beliefs in more propositions. This results in the following model:

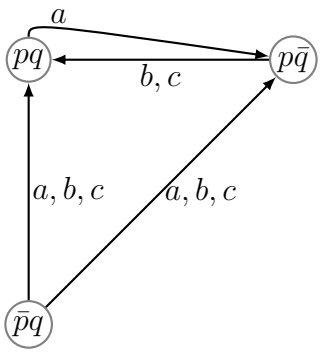


So a has the greatest belief in the world where p and not q hold, but after that she has more belief in a world where p and q both hold than in the world where q and not p hold, because in the first world at least her belief in p is right. Similarly for c. For b, she believes in the world where p and q hold, and values the other worlds equally plausible.

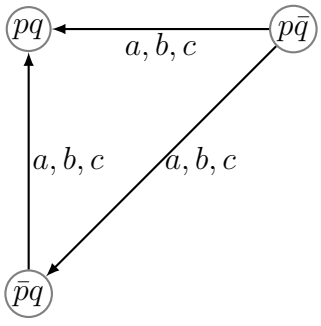
In this model the following formulas hold:

$$[\rightarrow_a]p, [\rightarrow_b]p, [\rightarrow_b]q, [\rightarrow_c]q, [\rightarrow_a]\neg(p \wedge q), [\rightarrow_c]\neg(p \wedge q).$$

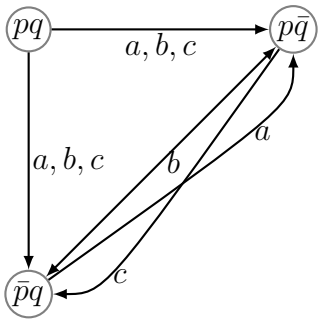
This shows that there are majority believes in p and in q, but there is also a majority belief in $\neg(p \wedge q)$. If the judges decide to have a Dutch meeting about p, the result will be unanimous belief in p:



Now if the judges hold a subsequent Dutch meeting about q, the result will be unanimous belief in q:



Now the judges unanimously believe in $p \wedge q$, so the defendant will be judged guilty. However, if a Dutch meeting about $p \wedge q$ was held in the first place, the result would be belief in $\neg(p \wedge q)$:



Clearly, in this case the defendant would be acquitted.

Experienced judges are of course familiar with this phenomenon. Procedural discussions about how to decompose a problem, and in which order to discuss the component problems may seem beside the point of a legal issue, but they turn out to be highly relevant for the outcome of the legal deliberations.

8.6 Conclusion

In this chapter I have studied the interplay between knowledge and belief. I have proposed a way to model knowledge and belief by using Kripke models with plausibility or preference relations. Unlike earlier approaches to modeling beliefs, I have not imposed strong requirements on the relations in my models. Instead, I have constructed modalities with the appropriate properties from unconstrained relations. This way I have shown how propositional dynamic logic with converse can be used as a basis for developing a very expressive system of multi-agent belief revision and belief change.

I have also studied the constraint for beliefs to be linked as a natural requirement for multi-agent belief change. Linkedness can be seen as a weaker version of local connectedness, extended to the multi-agent case. I have constructed an update mechanism that influences the belief of the agents while retaining the property of linkedness.

Since my logic provides a general mechanism for simultaneous belief change, it can be used to describe and analyze topics in judgement aggregation, the effects of agenda setting, the effects of subgroup meetings to create general belief, and many further issues of collective rationality.

Chapter 9

The Logic of Lying

9.1 Introduction

In the first part of this thesis I considered models of truthful communication. Furthermore, in Chapter 8 I considered a model of belief and belief revision, which can alternatively be viewed as a model of preference and preference aggregation. Here, I will investigate what happens when agents hear a lie, which they may believe or not. This chapter has a somewhat more philosophical flavour than the previous chapters, which are of a more technical nature.

The first question I would like to ask is the following: *What is a lie?*

The church father St. Augustine, who wrote at length about lying in *De Mendacio* [St. Augustine, 1988], holds a subtle view on what lying is and what it is not. I will take his view as our point of departure. Here is his famous quote on what lying is not.

> For not every one who says a false thing lies, if he believes or opines that to be true which he says. Now between believing and opining there is this difference, that sometimes he who believes feels that he does not know that which he believes, (although he may know himself to be ignorant of a thing, and yet have no doubt at all concerning it, if he most firmly believes it:) whereas he who opines, thinks he knows that which he does not know. Now whoever utters that which he holds in his mind either as belief or as opinion, even though it be false, he lies not. For this he owes to the faith of his utterance, that he thereby produce that which he holds in his mind, and has in that way in which he produces it. Not that he is without fault, although he lie not, if either he believes what he ought not to believe, or thinks he knows what he knows not, even though it should be true: for he accounts an unknown thing for a known.

> St. Augustine, *De Mendacio* (On Lying), ca. AD 395 [St. Augustine, 1988]

And on what lying is:

> Wherefore, that man lies, who has one thing in his mind and utters another in words, or by signs of whatever kind. Whence also the heart of him who lies is said to be double; that is, there is a double thought: the one, of that thing which he either knows or thinks to be true and does not produce; the other, of that thing which he produces instead thereof, knowing or thinking it to be false. Whence it comes to pass, that he may say a false thing and yet not lie, if he thinks it to be so as he says although it be not so; and, that he may say a true thing, and yet lie, if he thinks it to be false and utters it for true, although in reality it be so as he utters it. For from the sense of his own mind, not from the verity or falsity of the things themselves, is he to be judged to lie or not to lie. Therefore he who utters a false thing for a true, which however he opines to be true, may be called erring and rash: but he is not rightly said to lie; because he has not a double heart when he utters it, neither does he wish to deceive, but is deceived. But the fault of him who lies, is the desire of deceiving in the uttering of his mind; whether he do deceive, in that he is believed when uttering the false thing; or whether he do not deceive, either in that he is not believed, or in that he utters a true thing with will to deceive, which he does not think to be true: wherein being believed, he does not deceive though it was his will to deceive: except that he deceives in so far as he is thought to know or think as he utters.
>
> St. Augustine, [St. Augustine, 1988]

I cannot do better than to follow St. Augustine in assuming that the intention to mislead is part of the definition of a liar. Thus, to me, lying that p is communicating p in the belief that $\neg p$ is the case, with the intent to be believed.

The deceit involved in a lie that p is successful, if p is believed by the addressee after the speaker's utterance. This is my perspective. As is common in dynamic epistemic logic, I model the agents addressed by the lie, but I do not (necessarily) model the speaker as one of those agents. Dynamic epistemics model how to incorporate novel information *after* the decision to accept that information, just like in belief revision. I do not claim that this decision is irrelevant, far from that, but merely that this is a useful abstraction allowing me to focus on the information change only. This further simplifies the picture: I do not need to model the intention of the speaker, nor do I need to distinguish between knowledge and belief of the speaker: he is the observer of the system and his beliefs are taken to be the truth by the listeners. In other words, instead of having a precondition 'the speaker believes that p is false' for a lie, I have as a precondition 'p is false'.

In the previous chapters on truthful communication, the relations of the models I used were equivalence relations. In other words, the models were S5 models. In Chapter 8 I already briefly mentioned the fact that while truthful communication corresponds to S5 models, belief is often taken to correspond to KD45 models. I will now focus on these KD45 models. The logic also allows for even less specific notions than knowledge or belief. My analysis applies to all equally, and for all such epistemic notions I will use a doxastic modal operator $B_a p$, for 'agent a believes that p'. My analysis is not intended as a contribution to epistemology. I am aware of the philosophical difficulties with the treatment of knowledge as (justified) true belief [Gettier, 1963].

It is also possible to *model the speaker explicitly* in a modal logic of lying (and I will do so in examples) and extend my analysis to multi-agent systems wherein the deceptive interaction between speakers and hearers is explicit in that way. However, I do not explore that systematically here.

The *intention to be believed* can also be modeled in a (modal) logical language, namely by employing, for each agent, a preference relation that is independent from the accessibility relation for belief. This is to account for the fact that people can believe things for which they have no preference, and vice versa. This perspective is, e.g., employed in [Sakama et al., 2010] - this contains further references to the expansive literature on beliefs and intentions.

The *moral sides to the issue of lying* are clarified in the ninth of the ten commandments ('Thou shalt not bear false witness') and the fourth of the five Buddhist precepts ('I undertake the precept to refrain from false speech'). On the other hand, in the *Analects* of Confucius, Confucius is quoted as condoning a lie if its purpose is to preserve social structure:

> The Governor of She said to Confucius, 'In our village we have an example of a straight person. When the father stole a sheep, the son gave evidence against him.' Confucius answered, 'In our village those who are straight are quite different. Fathers cover up for their sons, and sons cover up for their fathers. In such behaviour is straightness to be found as a matter of course.' *Analects, 13.18.*

Among philosophical treatises, the quoted text of St. Augustine is a classic. For more, see [Bok, 1978] and [Arendt, 1967] and the references therein.

Rather than dwell on the moral side of the issue of lying, here I will study its logic, focusing on simple cases of lying in game situations, and on a particular kind of public announcement that may be deceptive and that I call 'manipulative update'. Thus, I abstract from the moral issues. I feel that it is important to understand why lying is tempting (why and how it pays off) before addressing the choice between condemnation and absolution.

The rest of the chapter is structured as follows. First, in Section 9.2, I develop a logic of lying in public discourse, treating a lie as an update with a communication believed to be truthful. Next, I turn to lying in games, by analyzing the game

of Liar's Dice, first in terms of game theory (Section 9.3), next in terms of (an implementation of) my logical system (Section 9.4). Section 9.5 concludes with a reflection on the difference between my logic of lying as manipulative update and lying in Liar's Dice.

9.2 The Logic of Lying in Public Discourse

We get lied to in the public domain, all the time, by people who have an interest in obfuscating the truth. In 1993 the tobacco company Philip Morris tried to discredit a report on *Respiratory Health Effects of Passive Smoking* by founding, through a hired intermediary, a fake citizen's group called *The Advancement of Sound Science* or TASSC, to cast doubt on it. Exxon-Mobile used the same organisation to spread disinformation about global warming.[1] Their main ploy: hang the label of 'junk science' on peer-reviewed scientific papers on smoking hazards or global warming, and promote propaganda disguised as research and 'sound science'. It worked beautifully for a while, until the *New York Times* exposed the fraud [Montague, April 29, 1998]. As a result, many educated people are still in doubt about the reality of global warming, or think the issues are just too hard for them to understand.

> It has frequently been noted that the surest result of brainwashing in the long run is a peculiar kind of cynicism, the absolute refusal to believe in the truth of anything, no matter how well it may be established. In other words, the result of a consistent and total substitution of lies for factual truth is not that the lie will now be accepted as truth, and truth be defamed as lie, but that the sense by which we take our bearings in the real world -and the category of truth versus falsehood is among the mental means to this end - is being destroyed.
>
> Hannah Arendt, "Truth and Politics", 1967 [Arendt, 1967].

Now this situation where complete cynicism reigns is one extreme attitude to confront lying. This is of course at the price of also no longer believing the truth. This attitude will be explored in my analysis of the game Liar's Dice, where the rules of the game allow any utterance regardless of its truth. The only thing that counts is winning. As everyone knows this, this is some kind of fair play.

The other extreme is the attitude where all lies are believed. This will be the logic of successful lies, where I take successful to mean that the addressees accept the lie as truth, even at the price of believing inconsistencies. Below I will give a logic of possibly deceptive public speech acts, to model the effects of lying as in politics. Proposition 9.2.10 below can be seen as a clear vindication that Arendt is right about the grave consequences of lying in politics.

[1] See `http://www.exxonsecrets.org/html/orgfactsheet.php?id=6`.

I will use Kripke models as defined in Chapter 2 to model the beliefs of a group of agents, and the modal language presented there to reason about them. I will use $B_a\phi$ as a shorthand for $[a]\phi$. It expresses that agent a believes ϕ. I will use action models with substitutions as defined in Chapter 3, Definition 3.3.4 to model the event that the agents hear a lie. The constraint I will put on these models is that they are KD45 models, as defined in Chapter 2. The class of KD45 models is characterized by the following axioms:

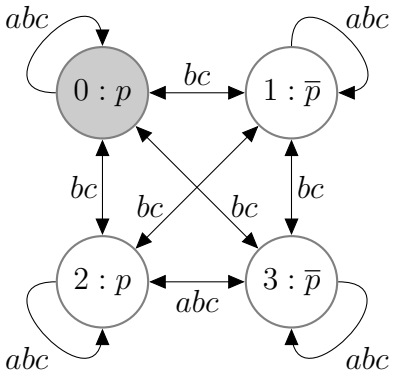
$$\begin{array}{lcl} & \neg B_a\bot & \\ B_a\phi & \rightarrow & B_aB_a\phi \\ \neg B_a\phi & \rightarrow & B_a\neg B_a\phi \end{array}$$

The first axiom states that no agent believes an inconsistency. The second is called ***positive introspection***, and it states that if an agent believes something, then he believes that he beliefs it. The third axiom is ***negative introspection***: if an agent does not believe something, then he believes that he does not believe it.

If I would also want to model the *intention* to deceive, I would need to use doxastic *preference* models (W, V, R, S), where S is a second relation for preference. Then it is reasonable to let S satisfy the KD45 postulates, or the constraint of linkedness that I presented in Chapter 8. But rather than carry such preference relations along in the exposition, I will indicate at appropriate places how they can be dealt with.

As I already indicated in Chapter 8 there is a problem with the logic of KD45 structures with KD45 updates, namely that this model class is not closed under execution of such updates. A single-agent example suffices to demonstrate this: consider a KD45 agent incorrectly believing that p: $\neg p \wedge B_i p$. Now inform this agent of the truth of $\neg p$. Then his accessibility relation becomes empty and is no longer serial. Another way to see that KD45 is no longer satisfied is by observing that the axiom $\neg B_a\bot$ no longer holds. The agent now believes everything! This means that the logic that incorporates updates with any action model as modal operators such as proposed in [van Benthem et al., 2006] cannot be complete with respect to the class of KD45 Kripke models. Therefore, I will not include a modal operator that consists of the update with an arbitrary action model in my logic. Rather, I will introduce certain updates representing a lie that will preserve the KD45 properties.

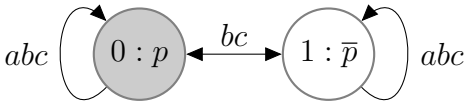
First, take the prototypical example of lying about p. Picture an initial situation where agent a knows that p, and agent a knows that agents b and c do not know that p. One way to picture this initial situation is like this:



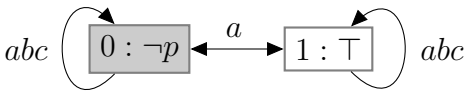
The gray shading indicates that 0 is the actual world. Because the relations are no longer assumed to be reflexive, in this chapter I will explicitly draw all reflexive relations. Note that agent a believes that p (agent a even *knows* that p, but this difference is immaterial to my analysis), but agents b, c also consider it possible that agent a believes the opposite (which is the case in world 1), or that agent a has no beliefs whatsoever about p (the situation in worlds 2 and 3).

In typical examples of bearing witness in court, the situation is often a bit different. In cases of providing an alibi, for example, the question 'Was the accused at home with you during the evening of June 6th?' is posed on the understanding that the witness is in a position to know the true answer, even if nobody can check that she is telling the truth.

Let us assume that everyone knows that a knows whether p. The picture now becomes:

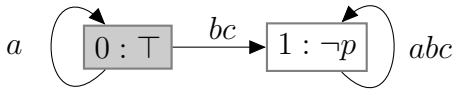


Assume agent a sends a group communication to b, c to the effect that $\neg p$. Would the following action model be a correct representation of the lie that $\neg p$?

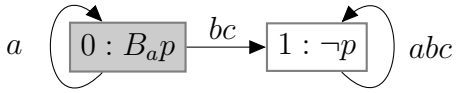


It is easy to see that this cannot be right. The result of this update is a model that has no actual worlds, i.e., an inconsistent model, since the actual world has p true, and the precondition of the actual action is $\neg p$.

Rather, the misleading communication should be modeled as a KD45 action model, as follows:



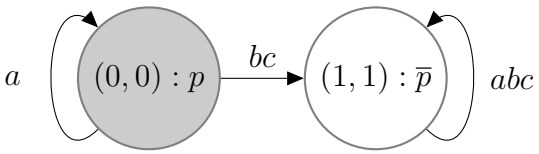
The misleading agent a knows that no truthful communication is being made, but the two agents b, c mistakenly believe that $\neg p$ is truthfully being asserted. The fact that the originator of the lie does believe that p is true can be taken on board as well, of course:



This update can equally be seen as agent a lying about p, or as an observer, not modeled in the system, lying about agent a believing that p. It cannot be called an explicit of a lie by agent a, because it cannot be distinguished from the (in fact more proper) perspective of an observer 'knowing' (believing, and with justification, as he is omniscient) that $B_a p$.

In the context of doxastic *preference* models, the precondition for the actual action could be extended even further, with the intent to mislead: in a's most preferred worlds, his victims believe that $\neg p$. I will omit the formal details in the interest of readability.

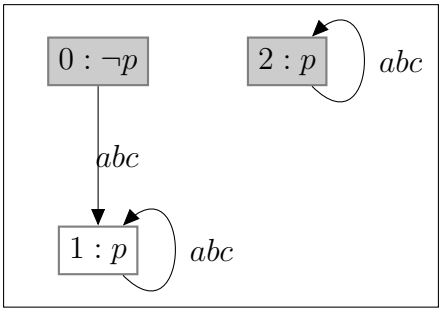
Updating the initial model with this action model gives:



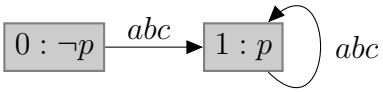
This is a model where a believes that p, where b, c mistakenly believe that $\neg p$, and where b, c also believe that a believes that $\neg p$. Note that the model is KD45: beliefs are still consistent ($[a]\phi \rightarrow \langle a \rangle \phi$ holds in the model), but the model is not truthful anymore (there are ϕ and a for which $[a]\phi \rightarrow \phi$ does not hold, i.e., there are false beliefs).

This way to model lying suggests a natural generalization of the well-studied concept of a public announcement. In the logic of public announcements [Plaza,

1989, Gerbrandy, 1999], a public announcement $!\phi$ is always taken to be a *true* statement. A more realistic version of public announcements leaves open the possibility of deceit, as follows. A possibly deceptive public announcement ϕ is a kind of 'if then else' action. In case ϕ is true, the announcement is a public update with ϕ, in case ϕ is false, the public is deceived into taking ϕ as true. The manipulative update with p by an outside observer (the announcer/speaker, who is not modeled as an agent in the structure), in a setting where the public consists of a, b, c, looks like this:



There are two actual events, one for the situation where p is true - in this case, the public is duly informed - and one for the situation where p is false - in this case the public is misled to believe that p. This action model can be simplified, as follows:



Call this the two-pointed manipulative update for p. I will refer to this action model as U_p. I will refer to the variation on this action model where only event 0 is actual as U_p^0. This action model denotes the lie with p. I will refer to the variant with only event 1 actual as U_p^1. This action model denotes the public announcement with p.

Let me introduce operations for these actions. The manipulative update with ϕ is denoted $\ddagger\phi$, and its two variants are denoted $¡\phi$ (for the lie that ϕ) and $!\phi$ (for the public announcement that ϕ).

I will include these updates as modal operators in my language. Define the logic of individual belief and manipulative update $\mathcal{L}_{BM}$ as follows:

$$\phi \quad ::= \quad p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid B_i\phi \mid [\ddagger\phi_1]\phi_2 \mid [¡\phi_1]\phi_2 \mid [!\phi_1]\phi_2$$

Interpretation as sketched above:

- $[\ddagger\phi]\psi$ is true in a model M at a world w if ψ is true in both $(w, 0)$ and $(w, 1)$ of the updated model $M \otimes U$.

- $[¡\phi]\psi$ is true in a model M at a world w if ψ is true in $(w, 0)$ of the updated model $M \otimes U^0$.

- $[!\phi]\psi$ is true in a model M at a world w if ψ is true in $(w, 1)$ of the updated model $M \otimes U^1$.

Now it turns out that the logic of individual belief and manipulative update has a simple axiomatisation in terms of reduction axioms, just like the logic of individual knowledge and public announcement. These reduction axioms are as follows. I start out with the reduction axioms for the $[\ddagger\phi]$ modality:

$$[\ddagger\phi]\psi \quad \leftrightarrow \quad [¡\phi]\psi \wedge [!\phi]\psi$$

This defines the effect of $[\ddagger\phi]$ in terms of those of $[!\phi]$ and $[¡\phi]$. Next, there are the usual reduction axioms for public announcement:

$$\begin{aligned}
[!\phi]p &\leftrightarrow \phi \to p \\
[!\phi]\neg\psi &\leftrightarrow \phi \to \neg[!\phi]\psi \\
[!\phi](\psi_1 \wedge \psi_2) &\leftrightarrow [!\phi]\psi_1 \wedge [!\phi]\psi_2 \\
[!\phi]B_i\psi &\leftrightarrow \phi \to B_i[!\phi]\psi
\end{aligned}$$

Finally, the reduction axioms for lying:

$$\begin{aligned}
[¡\phi]p &\leftrightarrow \neg\phi \to p \\
[¡\phi]\neg\psi &\leftrightarrow \neg\phi \to \neg[¡\phi]\psi \\
[¡\phi](\psi_1 \wedge \psi_2) &\leftrightarrow [¡\phi]\psi_1 \wedge [¡\phi]\psi_2 \\
[¡\phi]B_i\psi &\leftrightarrow \neg\phi \to B_i[!\phi]\psi
\end{aligned}$$

The final axiom of this list is the most interesting: it expresses that believing ψ after a lie that ϕ amounts to the belief that a public announcement of ϕ implies ψ, conditioned by $\neg\phi$.

Since all these axioms have the form of equivalences, completeness of the calculus of manipulation and individual belief follows from a reduction argument, as in the case of public announcements with individual knowledge. I refer to [van Benthem et al., 2006] for a general perspective on proving communication logics complete by means of reduction axioms.

9.2.1. Theorem. *The calculus of manipulation and individual belief is complete for the class of the (multi-)modal KD45 models.*

Another way to see that the logic is complete is by means of the observation that this is the special case of the Logic of Communication and Change (LCC, [van Benthem et al., 2006]) where updates are restricted to manipulations, announcements and lies, and where doxastic programs are restricted to individual accessibilities.

Interestingly, my logic of manipulation is closely related to the variation on public announcement that is used in [Gerbrandy, 2007, Kooi, 2007] (and going back to [Gerbrandy, 1999]) to analyze the 'surprise exam puzzle', where public announcement of ϕ is defined as an operation that restricts the doxastic alternatives of the agents to the worlds where ϕ is true, i.e., all relations to $\neg\phi$ worlds are destroyed. Using $\dagger\phi$ for this alternative announcement, the corresponding reduction axiom is $[\dagger\phi]B_i\psi \leftrightarrow B_i(\phi \rightarrow [\dagger\phi]\psi)$.

A forerunner of this logic is the analysis of suspicions and lies in [Baltag, 2002], which is further elaborated in [Baltag and Smets, 2008] and [van Ditmarsch, 2008]; the latter (actually a follow-up of the first version of the paper, [van Ditmarsch et al., 2012], on which this chapter was based) addresses more agency aspects in lying, such as the assumption that the addressee does not yet (firmly) believe the opposite of the lie - you don't want to be caught out as a liar!

At first sight, this alternative semantics for announcement takes me outside of the framework sketched above. However, if $\dagger\phi$ is an alternative announcement, then I have:

9.2.2. Proposition. *$M, w \models [\dagger\phi]\psi$ iff $M, w \models [\ddagger\phi]\psi$.*

Alternative announcement turns out to be the same as manipulative updating, and this analysis can be viewed as a decomposition of alternative announcement into public lying and (regular) public announcement.

Regular public announcements can be expressed in terms of manipulative updating:

9.2.3. Proposition. *$\vdash [!\phi]\psi \leftrightarrow (\phi \rightarrow [\ddagger\phi]\psi)$.*

The proof is by induction on ψ and is left to the reader.

The logic of public announcement and the logic of manipulation have the same expressive power: this follows from the fact that they both reduce to multimodal KD45. But note that the logic of manipulative updating has greater 'action expressivity' than the logic of public announcement: the logic of $[!\phi]$ has no means to express an operation mapping S5 models to KD45 models, and $[\ddagger\phi]$ is such an operation.

As an example of reasoning with the calculus, I use the axioms to show that a manipulative update followed by a belief is equivalent to a belief followed by the corresponding public announcement:

9.2.4. Proposition. *$\vdash [\ddagger\phi]B_i\psi \leftrightarrow B_i[!\phi]\psi$.*

PROOF.

$$\begin{array}{rcl} [\ddagger\phi]B_i\psi & \leftrightarrow & ([¡\phi]B_i\psi \wedge [!\phi]B_i\psi) \\ & \leftrightarrow & ((\neg\phi \rightarrow B_i[!\phi]\psi) \wedge (\phi \rightarrow B_i[!\phi]\psi)) \\ & \leftrightarrow & B_i[!\phi]\psi. \end{array}$$

□

An important difference between manipulative update and public announcement shows up when I work out the preconditions of inconsistency after an update. For public announcements I get:

9.2.5. PROPOSITION. $\vdash [!\phi]\bot \leftrightarrow \neg\phi$.

PROOF.

$$\begin{array}{rcl} [!\phi]\bot & \leftrightarrow & [!\phi](p \wedge \neg p) \\ & \leftrightarrow & ([!\phi]p \wedge [!\phi]\neg p) \\ & \leftrightarrow & ([!\phi]p \wedge (\phi \rightarrow \neg[!\phi]p)) \\ & \leftrightarrow & ((\phi \rightarrow p) \wedge (\phi \rightarrow \neg p)) \\ & \leftrightarrow & \neg\phi \end{array}$$

□

This shows that a public announcement with ϕ leads to an inconsistent state iff the negation of ϕ is true. Similarly, it is easy to work out that a public lie that ϕ leads to an inconsistency iff ϕ is true, i.e., I can derive

9.2.6. PROPOSITION. $\vdash [¡\phi]\bot \leftrightarrow \phi$.

Using these propositions I can work out the preconditions for inconsistency after a manipulative update:

9.2.7. PROPOSITION. $\vdash [\ddagger\phi]\bot \leftrightarrow \bot$.

PROOF.

$$\begin{array}{rcl} [\ddagger\phi] & \leftrightarrow & ([!\phi]\bot \wedge [¡\phi]\bot) \\ & \leftrightarrow & (\neg\phi \wedge \phi) \\ & \leftrightarrow & \bot \end{array}$$

□

This means that a manipulative update in a consistent state will never lead to inconsistency (although, of course, it may lead to an agent having an inconsistent set of beliefs, which is different).

The following proposition about public announcements can be proved by induction on ϕ. It shows that if one updates with an inconsistency, the resulting model is inconsistent:

9.2.8. Proposition. $\vdash [!\bot]\phi \leftrightarrow \top$.

In the case of manipulatively updating with an inconsistency, the result is not an inconsistent model, but a model where all accessibilities have vanished. In the particular case of the belief of agent a, this gives:

9.2.9. Proposition. $\vdash [\ddagger\bot]B_a\phi \leftrightarrow \top$.

Proof.

$$\begin{array}{lll} [\ddagger\bot]B_a\phi & \leftrightarrow & ([!\bot]B_a\phi \wedge [¡\bot]B_a\phi) \\ & \leftrightarrow & (\top \wedge B_a[!\bot]\phi) \\ & \leftrightarrow & B_a[!\bot]\phi \\ & \overset{\text{Prop 9.2.8}}{\leftrightarrow} & B_a\top \\ & \leftrightarrow & \top. \end{array}$$

□

After a manipulative update with an inconsistency, the public will no longer be able to distinguish what is false from what is true.

Finally, the following proposition spells out under what conditions our 'sense by which we take our bearings in the real world' is destroyed. This happens exactly when we are manipulated into accepting as truth what flatly contradicts our firm belief:

9.2.10. Proposition. $\vdash [\ddagger\phi]B_i\bot \leftrightarrow B_i\neg\phi$.

Proof.

$$\begin{array}{lll} [\ddagger\phi]B_i\bot & \leftrightarrow & ([!\phi]B_i\bot \wedge [¡\phi]B_i\bot) \\ & \leftrightarrow & ((\phi \rightarrow B_i[!\phi]\bot) \wedge (\neg\phi \rightarrow B_i[!\phi]\bot)) \\ & \leftrightarrow & ((\phi \rightarrow B_i\neg\phi) \wedge (\neg\phi \rightarrow B_i\neg\phi)) \\ & \leftrightarrow & B_i\neg\phi. \end{array}$$

□

I can generalize my logic to a full logic of manipulative updating, i.e., according to the full relational action description in the Logic of Communication and Change. For details, see Section 9.6.

In this section I have investigated the effect of lying in public discourse. In such a setting the agents assume that they are told the truth and in the event of a lie, the agents hearing the lie do not believe that the announcement is actually a lie. This causes them to believe a false thing. In Section 9.4 I will analyze lying in a different setting, where the agents are playing a game of Liar's Dice and following a game strategy. But first, I will give a game-theoretical analysis of the game to see how lying affects a game's outcome.

9.3 Liar's Dice — Game-Theoretical Analysis

In his later years as a saint, St. Augustine held the opinion that lying, even in jest, is wrong, but as the young and playful sinner that he was before his turn to seriousness he may well have enjoyed an occasional game of dice. I will examine a simplified version of two-person Liar's Dice, and show by means of a game-theoretical analysis that it is precisely the possibility of lying - using private information in order to mislead an opponent - that makes the game interesting.

In my simplified version of Liar's Dice, the die is replaced by a coin. A typical move of the game is tossing a coin and inspecting the result while keeping it hidden from the other player. Here is a description of what goes on, and what the options of the two players are.

- Players a and b both stake one euro: Player a bets on heads, Player b bets on tails.
- Player a tosses a coin under a cup and observes the outcome (heads or tails), while keeping it concealed from player b.
- Player a *announces* either $\aleph Head$ or $\aleph Tail$.
- If a announces $\aleph Tail$, then she simply loses her one euro to player b and game ends (for a bets on heads, so she announces defeat).
- If a announces $\aleph Head$, she adds one euro to the stake and the game continues.
- In response to $\aleph Head$, b either *passes* (gives up) or *challenges* "I don't believe that, you liar") and adds 1 euro to the stake.
- If b passes, a wins the stake, and the game ends.
- If b challenges, and the toss was heads, a wins the stake, otherwise b wins the stake. The game ends.

Player a has two information states: *Heads* and *Tails*, while player b has a single information state, for player b cannot distinguish the two possible outcomes of the toss. I will give a game-theoretic analysis of how player a can exploit her 'information advantage' to the utmost, and of how player b can react to minimize her losses, on the assumption that the procedure is repeated a large number of times. The following picture gives the extensive game form. The first move is made by Chance; this move gives the outcome of the coin toss. Then player a reacts, letting her move depend on the toss outcome. Finally, player b decides whether to pass or challenge. This decision does not depend on the coin toss; player b cannot distinguish the state where a announced $\aleph Head$ after seeing heads

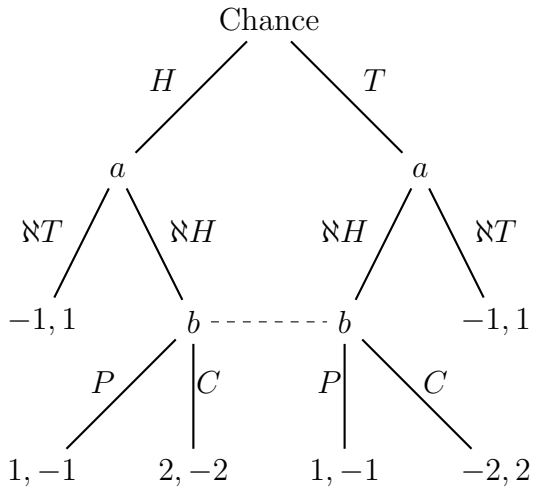


Figure 9.1: Extensive game form for Liar's Dice game.

from the state where she is bluffing. In the picture of the extensive game form (Figure 9.1) this is expressed by a dotted line.

The leaves of the game tree indicate the payoffs. If the game sequence is *Heads*, $\aleph Tail$, the payoffs are -1 euro for player a and 1 euro for player b. The same for the sequence *Tails*, $\aleph Tail$. Player a gets 1 euro and player b gets -1 euro for the sequences *Heads*, $\aleph Head$, *Pass*, and *Tail*, $\aleph Head$, *Pass* (these are the sequences where 2 gives up). The sequence *Heads*, $\aleph Head$, *Challenge* is a win for player a, with payoff 2 euros, and -2 euros for player b. The sequence *Tails*, $\aleph Head$, *Challenge*, finally, is a win for player b, with payoff 2 euros, and -2 euros for player a.

Player a has four strategies: *(*$\aleph Head$*,* $\aleph Head$*)* ($\aleph Head$ in case of heads and in case of tails), *(*$\aleph Head$*,* $\aleph Tail$*)* ($\aleph Head$ in case of heads, $\aleph Tail$ in case of tails), *(*$\aleph Tail$*,* $\aleph Head$*)*, and *(*$\aleph Tail$*,*$\aleph Tail$*)*. Player b has two strategies: *Pass* and *Challenge*. To find the strategic game form, one has to take the average of the expected payoffs for the two cases of *heads* and *tails*. E.g., if player a plays *(*$\aleph Head$*,* $\aleph Tail$*)* and player b responds with *Challenge*, then in the long run in $\frac{1}{2}$ of the cases the outcome will be heads, and player a wins 2 euros, and in $\frac{1}{2}$ of the cases the outcome will be tails, and player a loses 1 euro. Thus, the expected payoff is $\frac{1}{2} \times 2 - \frac{1}{2} \times 1 = \frac{1}{2}$ euro for player a, and because the game is zero sum, $-\frac{1}{2}$ euro for player b. The strategic game form is given by:

	Pass	Challenge
$\aleph Head$, $\aleph Head$	1,-1	0,0
$\aleph Head$, $\aleph Tail$	0,0	$\frac{1}{2}, -\frac{1}{2}$
$\aleph Tail$, $\aleph Head$	0,0	$-\frac{3}{2}, \frac{3}{2}$
$\aleph Tail$, $\aleph Tail$	-1,1	-1,1

It is easy to see that there is no pure strategy Nash equilibrium. A Nash equilibrium is a combination of strategies, one for each player, with the property that neither of the players can improve their payoff by unilaterally deviating from her strategy (see, e.g., [Osborne and Rubinstein, 1992]). Clearly, none of the eight strategy pairs has this property.

Now let's consider the strategy *(ℵTail, ℵTail)* for a. This is the strategy of the doomed loser: even when the toss is heads the player still announces $\aleph Tail$. This is obviously *not* the best thing that a can do. *Always* announcing $\aleph Head$ gives a much better payoff in the long run. In other words, the strategy *(ℵTail,ℵTail)* is strictly dominated by *(ℵHead, ℵHead)*. Similar for the strategy of the unconditional liar: *(ℵTail,ℵHead)*. It is also strictly dominated by the strategy *(ℵHead,ℵHead)*. Thus, I am left with:

	Pass	Challenge
$\aleph Head$,$\aleph Head$	1,-1	0,0
$\aleph Head$, $\aleph Tail$	0,0	$\frac{1}{2}, -\frac{1}{2}$

Suppose a plays *(ℵHead, ℵHead)* with probability p and *(ℵHead, ℵTail)* with probability $1-p$. Then her expected value is p for her first strategy, and $\frac{1}{2}(1-p)$ for her second strategy. Any choice of p where the expected payoff for p is different from that for $1-p$ can be exploited by the other player. Therefore, player a should play her first strategy with probability $p = \frac{1}{2}(1-p)$, i.e., $p = \frac{1}{3}$, and her second strategy with probability $1-p = \frac{2}{3}$. For player b, I can reason similarly. Suppose b plays *Pass* with probability q and *Challenge* with probability $1-q$. Again, the expected values for q and $1-q$ should be the same, for otherwise this mixed strategy can be exploited by the other player. The expected value is $-q$ for her first strategy and $-\frac{1}{2}(1-q)$ for her second strategy. Thus, she should play her first strategy with probability $q = \frac{1}{2}(1-q)$, i.e., $q = \frac{1}{3}$. Neither player can improve on her payoff by unilateral deviation from these strategies, so the mixed strategy where a plays *(ℵHead, ℵHead)* in $\frac{1}{3}$ of the cases and b plays *Pass* in $\frac{1}{3}$ of the cases is a Nash equilibrium. In other words, the best thing that player a can do is always announcing the truth and raising the stakes when her toss is heads, and lying in one third of the cases when her toss is tails, and b's best response to this is to Pass in one third of all cases and Challenge two thirds of the time.

The game-theoretic analysis yields that lying pays off for player a, and that player b, knowing this, may reasonably expect to catch player a on a lie in one sixth of all cases. The value of the game is $\frac{1}{3}$ euro, and the solution is $\frac{1}{3}$ ($\aleph Head$, $\aleph Head$), $\frac{2}{3}$ ($\aleph Head$, $\aleph Tail$) as player a's optimal strategy, and $\frac{1}{3}$ Pass, $\frac{2}{3}$ Challenge as player b's optimal strategy. It is clear that the honest strategy ($\aleph Head$, $\aleph Tail$) is not the optimal one for player a: given that player b plays $\frac{1}{3}$ Pass and $\frac{2}{3}$ Challenge, the expected payoff for player a is only $\frac{1}{6}$ if she sticks to the honest strategy. Lying indeed pays off sometimes.

If I modify the game so that player a cannot lie anymore, by refusing her the privilege of having a peek at the toss outcome, the game immediately becomes a lot less interesting. In the extensive game form for this version, an extra dotted line indicates that player a cannot distinguish the outcome *Heads* from the outcome *Tails*. See Figure 9.2.

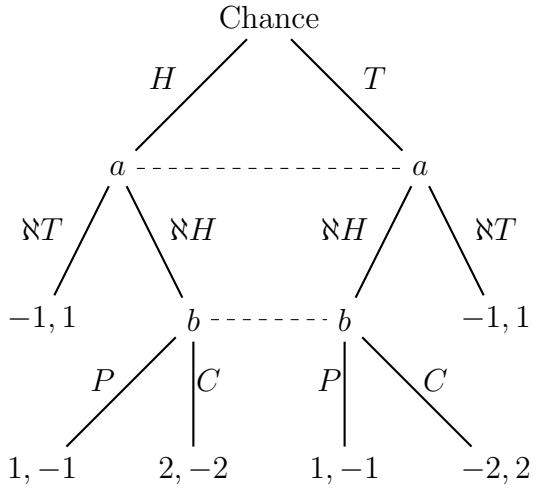


Figure 9.2: Modified game where player a has no information advantage.

Player a has just two strategies left, $\aleph Head$ and $\aleph Tail$, and the strategic form of the game becomes:

	Pass	Challenge
$\aleph Head$	1,-1	0,0
$\aleph Tail$	-1,1	-1,1

The strategy $\aleph Tail$ for player a is weakly dominated by $\aleph Head$, so it can be eliminated, and we are left with:

	Pass	Challenge
$\aleph Head$	1,-1	0,0

The strategy pair *($\aleph Head$, Challenge)* is a Nash equilibrium. The game-theoretic analysis predicts that a rational player a will always play $\aleph Head$, and a rational player b will always Challenge, and the game becomes a pure zero-sum game of chance. Surely, it is the possibility of lying that makes Liar's Dice an interesting game.

9.4 Liar's Dice — Doxastic Analysis

In the game of Liar's Dice, when player a announces *Heads* while she actually saw that the outcome of the toss was *Tails*, she is announcing something which

she believes to be false with the intent to be believed. This certainly seems to be a lie. However, we usually do not condemn people who tell such a lie in a game as untruthful. In fact, in this game player a is supposed to lie sometimes, or she would never win. This is an important point: player a *intends* player b to believe her, but she probably does not expect it, because player b may very well expect player a to lie sometimes. As I have already shown, it is completely immaterial in Liar's Dice whether an announcement is true or false: the only reasons for one or the other are strategic, and in view of winning the game. In this section I will analyze the game of Liar's Dice from a doxastic viewpoint in order to answer the question: is lying really lying, when one is actually *supposed* to lie?

For my analysis I will use the doxastic model checker DEMO [van Eijck, 2007]. Using DEMO, I can automatically check the truth of formulas in a doxastic model. I have extended DEMO with factual changes to allow action models with substitutions and also with the possibility to store integer values in my Bachelor's Thesis [Sietsma, 2007]. I will use this extended model checker. The code of this model checker is available from `http://www.cwi.nl/~jve/software/demolight0/`. I show how the game of Liar's Dice can be modeled using DEMO, and I demonstrate the doxastic models that I get if I trace a particular run of the game. For full details, see Section 9.7.

The conclusion of this analysis is that, even though in the game of Liar's Dice lying takes place according to the definition of Augustine, no misleading is taking place and the players are never duped into believing a falsehood. This is shown by the fact that all updates in the games, as modeled in the Appendix, are S5 updates: instead of unquestioningly taking for granted what they are being told, all players consider the opposite of what they are being told equally likely. In the resulting models there are no false beliefs, only true knowledge.

9.5 Conclusion

First of all, I will compare the approach presented here to that of Chapter 8. There, the only constraint on the basic relations is that they are linked and from these basic relations four different notions of belief are constructed using PDL. Here, all relations satisfy the KD45 axioms and I only use one notion of belief. The notion used here is probably closest to the notion of strong belief discussed there, although the relations in my model do not need to be reflexive while strong belief is constructed as the reflexive transitive closure of the basic relations. Using one single notion of belief allowed me to focus on the effects of lies on an agent's belief. The update discussed here differs from the one proposed in Chapter 8 because it results in "stronger" belief of the formula that is communicated. This is appropriate for the interpretation as a lie that is believed by the agents who hear it. In Chapter 8 the agents' relations represent preference or a "softer" form of belief, that allows for different levels of plausibility or preference. Such

an interpretation is more appropriate for the modeling of belief revision and judgement aggregation.

There are still two discrepancies that I have to address. The first one is between my treatment of lying in public discourse and my treatment of lying in games. As I have shown, lying in public discourse can lead to KD45 models, which illustrates the fact that genuine misleading takes place. I argued that the players in a game like Liar's Dice are never actually misled, so in a sense no real lying takes place here at all. But one might also say that lying is *attempted*, but due to the smartness of the opponent, these attempts are never really believed. So lying in public discourse and lying in games are connected after all.

The difference between the two settings could be seen as a difference in the *protocol* the agents are following. In public discourse, the agents usually assume that they are following the protocol "only speak the truth". Therefore, when one of them deviates from the protocol by telling a lie, the others believe him and are misled. In the game of Liar's Dice, the protocol is "say anything in order to improve your payoff". Since all agents know that the others are following the protocol, under the assumption of common knowledge of rationality, they do not believe each other's lies. The issue of protocol dynamics in epistemic modeling is explored further in [Wang, 2010].

The second discrepancy is between the game-theoretical analysis of lying in games in terms of mixed strategies that use probabilities, and the logical analysis in terms of truth values. To see that these perspectives still do not quite match, consider the game situation where player a tosses the coin, observes the result, and announces 'heads'. In my logical analysis this does *not* lead to the false belief of player b that the coin has landed heads; it does not lead to a belief change at all. But the game-theoretical analysis reveals that a rational agent would have formed a belief about the probability that the claim is true. So it seems that the logical analysis is still too crude.

This defect could be remedied by using probabilistic beliefs and probabilistic updates, in the style of [van Benthem et al., 2009b], which would allow me to express the probability of actions in the game. With these, one can model the fact that the game-theoretical analysis in terms of mixed strategies is common knowledge. For if this is the case, it is common knowledge that if the toss is tails, then player a will announce 'heads' with probability $\frac{1}{3}$ and 'tails' with probability $\frac{2}{3}$.

Interestingly, this is also relevant for the first discrepancy. For why are the players not duped into believing falsehoods, in the game of Liar's Dice? Because they look further than a single run of the game, and they know that as the game gets repeated they can adhere to mixed strategies. Therefore, an analysis in terms of manipulative probabilistic updates might work for both lying in public discourse and lying in games.

9.6 Appendix: The Full Logic of Manipulative Updating

The full logic of manipulative updating extends the logic of lies and individual beliefs from Section 9.2 to doxastic PDL. It consists of doxastic PDL extended with manipulative updates, lies and announcements:

$$\begin{array}{rcl} \alpha & ::= & i \mid ?\phi \mid \alpha_1;\alpha_2 \mid \alpha_1 \cup \alpha_2 \mid \alpha^* \\ \phi & ::= & p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid [\alpha]\phi \mid [\ddagger\phi_1]\phi_2 \mid [¡\phi_1]\phi_2 \mid [!\phi_1]\phi_2 \end{array}$$

There is a complete axiomatisation: the axioms and rules of PDL, the axioms of KD45, necessitation for $[\ddagger\phi]$, $[¡\phi]$, $[!\phi]$, and the following reduction axioms for the three update modalities.

The definition of $\ddagger$ in terms of ¡ and ! is as in Section 9.2:

$$[\ddagger\phi]\psi \quad \leftrightarrow \quad [¡\phi]\psi \wedge [!\phi]\psi$$

Reduction axioms for public announcement are as follows:

$$\begin{array}{rcl} [!\phi]p & \leftrightarrow & \phi \to p \\ [!\phi]\neg\psi & \leftrightarrow & \phi \to \neg[!\phi]\psi \\ [!\phi](\psi_1 \wedge \psi_2) & \leftrightarrow & [!\phi]\psi_1 \wedge [!\phi]\psi_2 \\ [!\phi][a]\psi & \leftrightarrow & [?\phi;a][!\phi]\psi \\ [!\phi][?\chi]\psi & \leftrightarrow & [?\phi;?\chi][!\phi]\psi \\ [!\phi][\alpha_1;\alpha_2]\psi & \leftrightarrow & [!\phi][\alpha_1][\alpha_2]\psi \\ [!\phi][\alpha_1 \cup \alpha_2]\psi & \leftrightarrow & [!\phi]([\alpha_1]\psi \wedge [\alpha_2]\psi) \\ [!\phi][\alpha^*]\psi & \leftrightarrow & [\alpha'^*][!\phi]\psi \\ & & \text{where } \alpha' \text{ such that } [!\phi][\alpha]\psi \leftrightarrow [\alpha'][!\phi]\psi \end{array}$$

It can be shown by an inductive argument that for every doxastic program α, every announcement $!\phi$, and every postcondition ψ a doxastic program α' exists such that $[!\phi][\alpha]\psi \leftrightarrow [\alpha'][!\phi]\psi$. This α', which does not have to be unique, can be found by applying the above reduction axioms.

Reduction axioms for public lies:

$$\begin{array}{rcl}
[¡\phi]p & \leftrightarrow & \neg\phi \to p \\
[¡\phi]\neg\psi & \leftrightarrow & \neg\phi \to \neg[¡\phi]\psi \\
[¡\phi](\psi_1 \wedge \psi_2) & \leftrightarrow & [¡\phi]\psi_1 \wedge [¡\phi]\psi_2 \\
[¡\phi][a]\psi & \leftrightarrow & [?\neg\phi; a][!\phi]\psi \\
[¡\phi][?\chi]\psi & \leftrightarrow & [?\neg\phi; ?\chi][!\phi]\psi \\
[¡\phi][\alpha_1;\alpha_2]\psi & \leftrightarrow & [¡\phi][\alpha_1][\alpha_2]\psi \\
[¡\phi][\alpha_1 \cup \alpha_2]\psi & \leftrightarrow & [¡\phi]([\alpha_1]\psi \wedge [\alpha_2]\psi) \\
[¡\phi][\alpha^*]\psi & \leftrightarrow & [\alpha'; \alpha''^*][!\phi]\psi \\
 & & \text{where } \alpha' \text{ such that } [¡\phi][\alpha]\psi \leftrightarrow [\alpha'][!\phi]\psi \\
 & & \text{and } \alpha'' \text{ such that } [!\phi][\alpha]\psi \leftrightarrow [\alpha''][!\phi]\psi
\end{array}$$

Again, it can be shown by an inductive argument that for every doxastic program α, every lie $¡\phi$, and every postcondition ψ, a doxastic programs α' exists such that $[¡\phi][\alpha]\psi \leftrightarrow [\alpha'][!\phi]\psi$.

The α' and α'' in the axioms for α^* can be viewed as the transformed versions of the programs α, where the update operator acts as a doxastic program transformer. To give an example, suppose $\alpha = a \cup b$, and I want to calculate the way common belief of a and b is transformed by a public lie that ϕ. Then the transformed program for $a \cup b$ becomes $?\neg\phi; a \cup b$, i.e., I have:

$$[¡\phi][a \cup b]\psi \leftrightarrow [?\neg\phi; a \cup b][!\phi]\psi.$$

Similarly for the way common belief of a and b is transformed by a public announcement: the transformed program for $a \cup b$ becomes $?\phi; a \cup b$, and I have:

$$[!\phi][a \cup b]\psi \leftrightarrow [?\phi; a \cup b][!\phi]\psi.$$

Using these transformed programs, one can see that the reduction axiom for $(a \cup b)^*$ takes the shape:

$$[¡\phi][(a \cup b)^*]\psi \leftrightarrow [?\neg\phi; a \cup b; (?\phi; a \cup b)^*][!\phi]\psi.$$

This expresses that after a lie with ϕ, a and b have a common belief that ψ iff in the model before the lie it holds that along all $a \cup b$ paths that start from a $\neg\phi$ world and that pass only through ϕ worlds, $[!\phi]\psi$ is true. Note that this is a 'relativized common belief' similar to the relativized common knowledge that is needed to get a reduction style analysis going of public announcement in the presence of common knowledge.

In fact, the style of axiomatisation that I have adopted is borrowed from the reduction axioms formulated in terms of program transformations, in [van Benthem et al., 2006]. In the same manner as in [van Benthem et al., 2006] I can derive (with the restriction to multi-K models, not to multi-KD45 models):

9.6.1. Theorem. *The calculus of manipulative updating is complete.*

9.7 Appendix: Liar's Dice in DEMO

First I will closely examine the different actions that take place in the game and their representations as action models. Let p represent the value of a coin, with 1 signifying heads, and 0 signifying tails. Let agents a and b represent the two players, and let C_1 represent the contents of the purse of player a (C for cash), and C_2 that of player b, with natural number values representing the amounts in euros that each player has in her purse. These natural number registers are available in the new extension of DEMO that was presented in [Sietsma, 2007]. Let S_1, S_2 represent the money at stake for each player. Factual change can be thought of as assignment of new values to variables. This is an essential ingredient of the various actions in the game:

Initialisation Both players put one euro at stake, and they both know this. $S_1 := 1, C_1 := C_1 - 1, S_2 := 1, C_2 := C_2 - 1$, together with public announcement of these factual changes.

Heads Factual change of the propositional value of a coin p to 1, with private communication of the result to player a ($p = 1$ signifies heads).

Tails Factual change of the propositional value of a coin p to 0, with private communication of the result to player a. ($p = 0$ signifies tails).

Announce Player a announces either $\aleph Head$ or $\aleph Tail$. There are several ways to model this and I will come back to this later.

Pass Player b passes and loses, player a gets the stakes. $C_1 := C_1 + S_1 + S_2, S_1 := 0, S_2 := 0$.

Challenge Public setting of $C_2 := C_2 - 1, S_2 := S_2 + 1$, followed by public announcement of the value of p. If the outcome is p then $C_1 := C_1 + S_1 + S_2$, otherwise $C_2 := C_2 + S_1 + S_2$ and in any case $S_1 := 0, S_2 := 0$.

I will show how these actions can be defined as doxastic action models in Haskell code using DEMO.

```
module Lies
where
import ModelsVocab hiding (m0)
import ActionVocab hiding (upd,public,preconditions,
                           vocProp,vocReg)
import ChangeVocab
import ChangePerception
import Data.Set (Set)
import qualified Data.Set as Set
```

```
type EM = EpistM Integer
```

I first define the cash and stakes of each player as integer registers.

```
c1, c2, s1, s2 :: Reg
c1 = (Rg 1); c2 = (Rg 2)
s1 = (Rg 3); s2 = (Rg 4)
```

This declares four integer registers, and gives them appropriate names. The initial contents of the purses of the two players must also be defined. Let us assume both players have five euros in cash to start with.

```
initCash1, initCash2 :: Int
initCash1 = 5
initCash2 = 5
```

Initialisation of the game: both players put one euro at stake. This is modeled by the following factual change: $S_1 := 1, C_1 := C_1 - 1, S_2 := 1, C_2 := C_2 - 1$. Representating this in my modeling language is straightforward. I just represent the contents of the registers at startup.

```
initGame :: EM
initGame = (Mo
            [0]
            [a,b]
    []
    [s1, s2, c1, c2]
    [(0,[])]
            [(0,[(s1,1),(s2,1),
                 (c1,(initCash1-1)),(c2,(initCash2-1))])]
            [(a,0,0),(b,0,0)]
            [0])
```

Tossing the coin is a factual change of p to 0 or 1. The coin is tossed secretly and before player a looks both players are unaware of the value of the coin. Therefore there are two worlds, one where p is set to 0 and one where p is set to 1, and neither of the two players can distinguish these worlds.

```
toss :: Integer -> FACM State
toss c ags = (Acm
              [0,1]
              ags
              [(0,(Top,([(P 0,Neg Top)],[]))),
               (1,(Top,([(P 0,Top)],[])))]
              [(ag,w,w') | w  <- [0,1],
                           w' <- [0,1], ag <- ags]
              [c])
```

Note that the action model has a list that assigns to each world a precondition, a change to the propositions, and a change to the registers. In world 0, the precondition is $\top$ and the change is to set p to value $\neg\top$, i.e., $\bot$ (and there is no change to the registers), and in world 1, the precondition is again $\top$ and the change is to set p to value $\top$ (and again, there is no change to the registers).

After the coin has been tossed player a looks under the cup without showing the coin to player b. I define a generic function for computing the model of the action where a group of agents looks under the cup. These models consist of two worlds, one where p is true (heads) and one where p is false (tails), the agents in the group can distinguish these two worlds and the other agents cannot.

```
look :: [Agent] -> FACM State
look group ags = (Acm
                  [0,1]
                  ags
                  [(0,(p,([],[]))),(1,(Neg(p),([],[])))]
                  ([(ag,w,w') | w <- [0,1], w' <- [0,1],
                         ag <- ags, notElem ag group] ++
                   [(ag,w,w)  | w <- [0,1], ag <- group])
                  [0,1])
```

In this case, there are no changes to propositions or registers, but world 0 has precondition p, and world 1 has precondition $\neg p$.

Now I define the models of the situation after the coin has been tossed and player a has looked at the outcome, distinguishing the two outcomes of the toss:

```
headsg :: EM
headsg = upd (upd initGame (toss 1)) (look [a])

tailsg :: EM
tailsg = upd (upd initGame (toss 0)) (look [a])
```

Before looking at the way to model the announcement of an outcome of the toss by player a I will first define the action models for passing and challenging.

When player b passes, the stakes are added to player a's cash: $C_2 := C_2 + S_1 + S_1, S_1 := 0, S_2 := 0$. Player b never gets to see the actual value of the coin so there are no changes in the knowledge of the agents about p. The model for this has only one world that indicates the changes in the stakes and cash.

```
pass :: FACM State
pass ags = (Acm
            [0]
            ags
            [(0,(Top,([],
                  [(s1,(I 0)),
                   (s2,(I 0)),
                   (c1,ASum [Reg c1,Reg s1,Reg s2])])))]
            [(ag,0,0) | ag <- ags]
            [0])
```

Note that here for the first time there are changes of the registers.

When player b decides to challenge player a, the cup is lifted and both players get to know the value of p. Then the stakes are added to the cash of player a in case of heads and player b in case of tails, together with one extra euro from the cash of player b that player b added to the stakes while challenging player a. So instead of $S2 := S2+1, C2 := C2-1$ and after that $C1 := C1+S1+S2$ in case of heads and $C2 := C2+S1+S2$ in case of tails, I use $C1 := C1+S1+S2+1, C2 := C2-1$ in case of heads and $C2 := C2 + S1 + S2$ in case of tails. The action model for this has one world for the case of heads and one world for the case of tails. Both players can distinguish these worlds because the cup was lifted, and the stakes are divided differently in the two worlds.

```
challenge :: FACM State
challenge ags =
 Acm
 [0,1]
 ags
 [(0,(Neg(p),([],
       [(s1,(I 0)),
        (s2,(I 0)),
        (c2,ASum [Reg c2,Reg s1,Reg s2])]))),
  (1,(    p ,([],
       [(s1,(I 0)),
        (s2,(I 0)),
        (c2,ASum [Reg c2,I (-1)]),
        (c1,ASum [Reg c1,Reg s1,Reg s2,I 1])])))]
  [(ag,w,w) | w <- [0,1], ag <- ags]
  [0,1]
```

When player a announces $\aleph Head$ or $\aleph Tail$ the stakes change. In case of $\aleph Head$ $C_1 := C_1 - 1, S_1 := S_1 + 1$ and in case of $\aleph Tail$ $C_2 := C_2 + S_1 + S_2, S_1 := 0, S_2 := 0$.

```
announceStakes :: Integer -> FACM State
announceStakes 0 ags =
 Acm
 [0]
 ags
 [(0,(Top,([],[(s1,(I 0)),
  (s2,(I 0)),
  (c2,ASum [Reg c2,Reg s1,Reg s2])])))]
 [(ag,0,0) | ag <- ags]
 [0]
announceStakes 1 ags =
 Acm
 [0]
 ags
 [(0,(Top,([],[(s1,ASum [Reg s1,I 1]),
  (c1,ASum [Reg c1,I (-1)])])))]
 [(ag,0,0) | ag <- ags]
 [0]
```

Now the only thing I have to decide is how I will model the announcement of $\aleph Head$ or $\aleph Tail$. Suppose I would use the manipulative update $\ddagger p$ or $\ddagger \neg p$ for this. This would imply that the other player believes the claims that are made.

I first define a generic function that computes the model for any manipulative update. This is the model with two worlds, one where the formula that is announced is true and one where it is false, and relations from the world where it is false to the world where it is true for the agents that believe the announcement.

```
manipulative :: Form -> [Agent] -> FACM State
manipulative f group ags =
 (Acm
  [0,1]
  ags
  [(0,(Neg f,([],[]))),(1,(f,([],[])))]
  ([(ag,w,w') | w <- [0,1], w' <- [0,1],
                ag <- ags, notElem ag group ] ++
   [(ag,w,1 ) | w <- [0,1], ag <- group ])
  [0,1])
```

Now when player a announces $\aleph Head$ or $\aleph Tail$ two things happen: the manipulative update is made to player 2, and player 1 adds one euro to the stakes in case of $\aleph Head$ or player b wins the stakes in case of $\aleph Tail$. I first model the manipulative update. In case of announcement of $\aleph Head$ this is the manipulative update with p, otherwise it is the manipulative update with $\neg p$.

```
announceManip :: Integer -> FACM State
announceManip c = manipulative (fct c) [b]
                  where fct 0 = Neg (Prp (P 0))
                        fct 1 = (Prp (P 0))
```

I can combine these action models in a function on doxastic models:

```
announce' :: Integer -> EM -> EM
announce' c m =
   upd (upd m (announceManip c)) (announceStakes c)
```

Now I have a complete way to model any game of Liar's Dice. However, though this way to model things seems correct, it is not. When I model player

a's announcement with manipulative updates player b will actually believe player a's announcement. I can use the model checker to show this:

```
*Lies> isTrue (announce' 0 headsg) (K b (Neg p))
True
*Lies> isTrue (announce' 0 tailsg) (K b (Neg p))
True
*Lies> isTrue (announce' 1 headsg) (K b p)
True
*Lies> isTrue (announce' 1 tailsg) (K b p)
True
```

However, in a real game of Liar's Dice player b knows that player a might very well be bluffing and she does not really believe player a's claim at all. So to correctly model the game I should not use the manipulative update. When player a makes an announcement this does not even change player b's knowledge and beliefs because player b does not believe player a.

So instead of the manipulative update I should only use the model for changing the stakes to model the announcement:

```
announce :: Integer -> FACM State
announce = announceStakes
```

Now player b does not know whether p is true, but she knows she doesn't know:

```
bKnows :: Form
bKnows = Disj [(K b (Neg p)), (K b p)]
```

```
*Lies> isTrue (upd tailsg (announce 0)) bKnows
False
*Lies> isTrue (upd tailsg (announce 0)) (K b (Neg bKnows))
True
*Lies> isTrue (upd headsg (announce 0)) bKnows
False
*Lies> isTrue (upd headsg (announce 0)) (K b (Neg bKnows))
True
*Lies> isTrue (upd tailsg (announce 1)) bKnows
False
*Lies> isTrue (upd tailsg (announce 1)) (K b (Neg bKnows))
```

```
True
*Lies> isTrue (upd headsg (announce 1)) bKnows
False
*Lies> isTrue (upd headsg (announce 1)) (K b (Neg bKnows))
True
```

Note that since I did not use the manipulative update to model player a's announcement the resulting models are still S5-models.

```
Lies> isS5Model (upd headsg (announce 1))
True
Lies> isS5Model (upd headsg (announce 0))
True
Lies> isS5Model (upd tailsg (announce 1))
True
Lies> isS5Model (upd tailsg (announce 0))
True
```

This means that no actual misleading is taking place at all! This is actually very plausible because player b knows that player a's announcement might very well be false. This shows that lying only creates false belief if the person who lies is believed to be telling the truth.

Now I can use these action models to do a doxastic analysis of a game of Liar's Dice. The different possible games are:

1. Player a tosses tails and announces $\aleph Tail$
2. Player a tosses heads and announces $\aleph Tail$
3. Player a tosses tails and announces $\aleph Head$ and player b passes
4. Player a tosses tails and announces $\aleph Head$ and player b challenges
5. Player a tosses heads and announces $\aleph Head$ and player b passes
6. Player a tosses heads and announces $\aleph Head$ and player b challenges

The models for these games are:

```
game1, game2, game3, game4, game5, game6 :: EM
game1 = gsm (upd tailsg (announce 0))
game2 = gsm (upd headsg (announce 0))
game3 = gsm (upd (upd tailsg (announce 1)) pass)
game4 = gsm (upd (upd tailsg (announce 1)) challenge)
game5 = gsm (upd (upd headsg (announce 1)) pass)
game6 = gsm (upd (upd headsg (announce 1)) challenge)
```

I will now consider these six different cases in turn.

Game 1 is the game where player 1 tosses tails and admits this.

In this case both players stake one euro and player b wins the stakes, so in the end player a lost one euro and player b won one euro. This can be checked with DEMO:

```
*Lies> isTrue game1 (Eq (Reg c1) (ASum [I initCash1,I (-1)]))
True
*Lies> isTrue game1 (Eq (Reg c2) (ASum [I initCash2,I 1]))
True
```

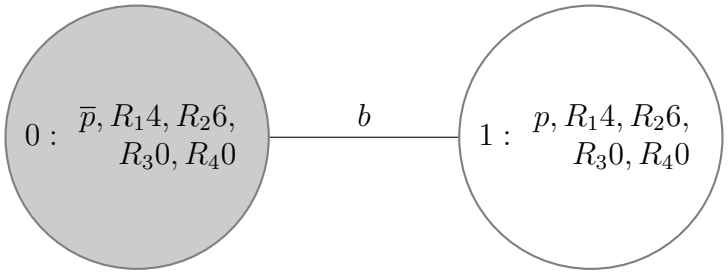
Player b does not get to know what the value of the coin was:

```
*Lies> isTrue game1 bKnows
False
```

The model for game 1 is:

```
*Lies> displayS5 game1
[0,1]
[p]
[R1,R2,R3,R4]
[(0,[]),(1,[p])]
[(0,[(R1,4),(R2,6),(R3,0),(R4,0)]),
 (1,[(R1,4),(R2,6),(R3,0),(R4,0)])]
(a,[[0],[1]])
(b,[[0,1]])
[0]
```

A picture of this model is below. There are two worlds, one where the toss was heads and one where it was tails. Player a can distinguish these worlds, but player b cannot because player b never got to see the coin. In both worlds the cash of player a is 4 and that of player b is 6 euros, because the division of the stakes does not depend on the value of the coin. Reflexive arrows are not shown.



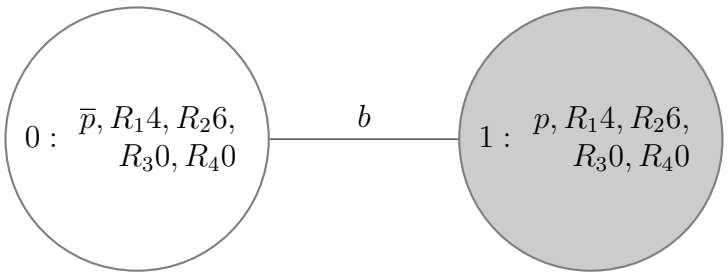
Game 2 is the game where player a falsely announces $\aleph Head$. Just like in game 1, player a loses one euro and player b wins one euro, and player b does not get to know the value of the coin.

```
*Lies> isTrue game2 (Eq (Reg c1) (ASum [I initCash1,I (-1)]))
True
*Lies> isTrue game2 (Eq (Reg c2) (ASum [I initCash2,I 1]))
True
*Lies> isTrue game2 bKnows
False
```

The model for this game is almost the same as for game 1: the difference is that now the world where p is true is actual instead of the world where p is false.

```
*Lies> displayS5 game2
[0,1]
[p]
[R1,R2,R3,R4]
[(0,[]),(1,[p])]
[(0,[(R1,4),(R2,6),(R3,0),(R4,0)]),
 (1,[(R1,4),(R2,6),(R3,0),(R4,0)])]
(a,[[0],[1]])
(b,[[0,1]])
[1]
```

The picture of this model (reflexive arrows not shown) is:



The third game is the case where player a tosses tails but falsely announces $\aleph Head$ and player b passes. In this case player a stakes two euros and player b stakes one euro, and player a gets to keep the stakes, so the final payoff is that player a wins one euro and player b loses one euro:

```
*Lies> isTrue game3 (Eq (Reg c1) (ASum [I initCash1,I 1]))
True
*Lies> isTrue game3 (Eq (Reg c1) (ASum [I initCash1,I 1]))
True
```

Player b passes, so the cup is never lifted and player b does not know the value of the coin:

```
*Lies> isTrue game3 bKnows
False
```

The model for this game is:

```
*Lies> displayS5 game3
[0,1]
[p]
[R1,R2,R3,R4]
[(0,[]),(1,[p])]
[(0,[(R1,6),(R2,4),(R3,0),(R4,0)]),
 (1,[(R1,6),(R2,4),(R3,0),(R4,0)])]
(a,[[0],[1]])
(b,[[0,1]])
[0]
```

This model has the same two worlds as the models for game 1 and 2 except for the changes in the player's cash.

In the fourth game, player a tosses tails but falsely announces $\aleph Head$ and player b challenges player a. This means that both players stake one extra euro and then the cup is lifted and player b gets the stakes.

In this case player b does know the value of the coin:

```
*Lies> isTrue game4 bKnows
True
```

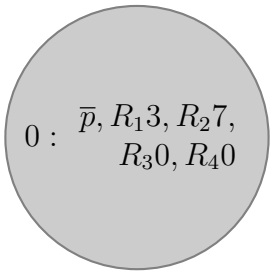
The payoffs are -2 euros for player a and 2 euros for player b:

```
*Lies> isTrue game4 (Eq (Reg c1) (ASum [I initCash1,I (-2)]))
True
*Lies> isTrue game4 (Eq (Reg c1) (ASum [I initCash1,I (-2)]))
True
```

The model for this game is:

```
*Lies> displayS5 game4
[0]
[p]
[R1,R2,R3,R4]
[(0,[])]
[(0,[(R1,3),(R2,7),(R3,0),(R4,0)])]
(a,[[0]])
(b,[[0]])
[0]
```

This model has only one world because none of the players consider any other world possible. This is because both players know the values of the coin. In this world p is false (because the toss was tails), player a's cash is 3 euros and player b's cash is 7 euros. A picture of this model is below.



The fifth game is the game where player a tosses heads and truthfully announces this and player b passes. In this case the cup is not lifted so player b does not know the value of the coin again:

```
*Lies> isTrue game5 bKnows
False
```

The payoffs are 1 for player a and -1 for player b:

```
*Lies> isTrue game5 (Eq (Reg c1) (ASum [I initCash1,I 1]))
True
*Lies> isTrue game5 (Eq (Reg c2) (ASum [I initCash2,I (-1)]))
True
```

The model for game 5 has two worlds again because player b does not know the value of the coin.

```
*Lies> displayS5 game5
[0,1]
[p]
[R1,R2,R3,R4]
[(0,[]),(1,[p])]
[(0,[(R1,6),(R2,4),(R3,0),(R4,0)]),
 (1,[(R1,6),(R2,4),(R3,0),(R4,0)])]
(a,[[0],[1]])
(b,[[0,1]])
[1]
```

In game 6 player a tosses heads and truthfully announces this and player b challenges player a. In this case both players add one extra euro to the stakes, the cup is lifted and player a gets to keep the stakes. The model for this has one world where p is true, player a has 7 euros and player b has 3 euros.

```
*Lies> displayS5 game6
[0]
[p]
[R1,R2,R3,R4]
```

```
[(0,[p])]
[(0,[(R1,7),(R2,3),(R3,0),(R4,0)])]
(a,[[0]])
(b,[[0]])
[0]
```

In this case player b knows the value of the coin and the payoffs are 2 euros for player 1 and -2 euros for player 2:

```
*Lies> isTrue game6 bKnows
True
*Lies> isTrue game6 (Eq (Reg c1) (ASum [I initCash1,I 2]))
True
*Lies> isTrue game6 (Eq (Reg c2) (ASum [I initCash2,I (-2)]))
True
```

Chapter 10

Conclusion

In this thesis I have studied the evolution of knowledge during communication between agents from a logical viewpoint. The great number of different perspectives I take in the different chapters show that there are many forms of communication. I mostly focussed on one-way communication through messages but even within this framework there are a lot of differences. This becomes very clear in Chapter 4. There, I give a very general approach in which many forms of communication can be modeled by adapting the model to the needs of the situation at hand. Several types of communicative actions can be defined, each with its own parameters, and every combination of parameters gives its own results in terms of knowledge evolution. I also give a clear definition of the network over which the agents communicate. The network can even be changed during the process of communication with a special action. It would be an interesting line of future research to see how this communication network can be incorporated in the approaches presented in the other chapters, which are more tailored to specific forms of communication. For example, in Chapters 5 and 6, which focus on email communication specifically, one could imagine the existence of certain "mailing lists" through which certain groups of agents can receive one shared email, while other agents can only be reached individually. Also, some agents may not know the email address of other agents, preventing them from contacting these agents directly. Then they might send their email to some third agent of which they do have the email address so this third agent can forward the message to the intended recipient.

Another potential topic of further work is to combine the concept of common knowledge discussed in Chapter 5 with the concepts of potential and definitive knowledge from Chapter 6. Such a study could start out with interpreting common knowledge under the assumption that everyone reads their messages immediately to arrive at "possible common knowledge" or under the assumption that everyone has only read email that they replied to in order to define "definitive common knowledge". But more complicated extensions are also possible, for ex-

ample one where the "reading behaviour" varies between agents. Then one could assume that there is one group of agents who always reads their email, and another group who can only be counted upon to have read emails they replied to. This could even lead to nested expressions like "it is possible common knowledge in group A that it is definitive common knowledge in group B that this message was sent". Continuing this line of thought, another interesting extension would be to investigate more kinds of reading behaviour than just "read everything immediately" or "read only what you reply to".

It is also promising to investigate whether one could extend the contents of the messages discussed in Chapters 6 and 5 to formulas rather than basic notes. This can be extremely powerful, especially if these formulas also contain epistemic operators. Then the agents could send each other emails containing information like "Alice knows about this message, but Bob does not know she knows it". It would require an intricate system of processing new information received by the agents. Such an approach would essentially combine and extend the strengths of Chapters 6 and 5 on the one hand and Chapter 3 on the other. In that chapter, the messages do contain formulas. These formulas do not contain epistemic operators, but because they can contain previous messages the language is already quite expressive. However, the downside of this approach is that the number of messages available to the agents must be limited to a finite set, which makes the set-up less general. It is still very suitable for many applications where a fixed protocol is being followed and it is also very relevant to many topics in game theory. If the limitation on the possible messages would be lifted this would result in a model of infinite size. This is essentially what happens in Chapter 5, where the complete model of all possible states is indeed infinite and therefore not represented explicitly. The model presented there is still a very nice theoretical representation, which allows for logical reasoning about the knowledge of the agents, in particular the common knowledge of a group of agents. However, I have not found a decision procedure for that model. This open question is solved for the framework presented in Chapter 6. There, the number of possible states is still infinite, but I have found a limit on the states that need to be evaluated in order to determine whether an agent knows something. This is a good solution for the problem of the infinite number of states. However, a finite model would allow for a better representation of the models in a way that is easy to understand for humans.

Another important open question concerns the work presented in Chapter 7. There, I present a notion of action emulation which is a relation between action models, meant to characterize their equivalence. For canonical action models, it does. For non-canonical action models, action emulation implies equivalence but it is yet unclear whether the converse is also true. Therefore, the open question is: does action model equivalence imply action emulation for non-canonical action models? If this holds then the notion of action emulation I presented is truly a new standard for action model equivalence. So far, I have found neither a proof

nor a counterexample.

In Chapter 8 I have studied the difference between knowledge and belief. I showed how knowledge relations in a model such as the ones used in Chapter 3 can be adapted to belief relations, and what consequences this has on the conditions we should impose on these relations. I also propose a new condition, that leads to the possibility to model a number of different kinds of belief. It would be interesting to combine this with the approach from Chapter 3 to a logic of messages and belief. One way to do this would be to give every message some "level of credibility" that determines how strongly the other agents believe its contents. This level of credibility might vary between the different agents depending on how prone they are to believe the message. It would be a big next step in epistemic logic to use a quantitative approach here, allowing one to compute for every agent the probability he gives to every possible event.

Such an approach would also be very relevant to Chapter 9, where I study the logic of lying. In this chapter I show how the act of telling a lie can be modeled as the manipulative update of an epistemic model. Furthermore, I study a game of Liar's Dice where the players may either speak the truth or lie as a part of their strategy to win the game. Probabilities play a big role there because both opponents want to maximize their expected profit after a number of rounds of the game. Therefore, a probabilistic approach is indeed very promising.

Bibliography

Thomas Ågotnes, Philippe Balbiani, Hans van Ditmarsch, and Pablo Seban. Group announcement logic. *Journal of Applied Logic*, 8:62–81, July 2009. Cited on page **54**.

E. A. Akkoyunlu, K. Ekanadham, and R. V. Huber. Some constraints and tradeoffs in the design of network communications. In *Proceedings of the Fifth ACM Symposium on Operating Systems Principles*, pages 67–74, 1975. Cited on page **2**.

Krzysztof R. Apt, Andreas Witzel, and Jonathan A. Zvesper. Common knowledge in interaction structures. In *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 4–13, 2009. Cited on pages **40**, **41**, **45**, **46**, and **61**.

H. Arendt. Truth and politics. In *Past and Future - Six Exercises in Political Thought*. Viking Press, 1967. Penguin Classics Edition, 2006. Cited on pages **153** and **154**.

László Babai. E-mail and the unexpected power of interaction. In *Fifth Annual Structure in Complexity Theory Conference: Proceedings*, pages 30–44, 1990. Cited on page **58**.

Alexandru Baltag. A logic for suspicious players: Epistemic action and belief-updates in games. *Bulletin of Economic Research*, 54(1):1–45, 2002. Cited on page **160**.

Alexandru Baltag and Lawrence S. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004. Cited on pages **37**, **38**, and **40**.

Alexandru Baltag and Sonja Smets. Conditional doxastic models: A qualitative approach to dynamic belief revision. *Electronic Notes in Theoretical Computer Science (ENTCS)*, 165:5–21, 2006. Cited on page **140**.

Alexandru Baltag and Sonja Smets. A qualitative theory of dynamic interactive belief revision. In *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, volume 3 of *Texts in Logic and Games*, pages 9–58. Amsterdam University Press, 2008. Cited on pages **136**, **140**, and **160**.

Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the 7th conference on Theoretical Aspects of Rationality and Knowledge (TARK '98)*, pages 43–56, 1998. Cited on pages **14** and **113**.

A. Baskar, R. Ramanujam, and S.P. Suresh. Knowledge-based modelling of voting protocols. In *Proceedings of Theoretical Aspects of Rationality and Knowledge*, pages 62–71, 2007. Cited on page **46**.

Ido Ben-Zvi and Yoram Moses. Beyond Lamport's *Happened-Before*: On the role of time bounds in synchronous systems. In *Proceedings of the 24th International Conference on Distributed Computing*, pages 421–436, 2010. Cited on page **60**.

Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001. Cited on pages **13**, **113**, and **118**.

Oliver Board. Dynamic interactive epistemology. *Games and Economic Behaviour*, 49(1):49–80, 2002. Cited on page **140**.

Sissela Bok. *Lying - Moral Choice in Public and Private Life*. The Harvester Press, 1978. Cited on page **153**.

Craig Boutilier. Toward a logic for qualitative decision theory. In *Proceedings of the 4th International Conference on Principle of Knowledge Representation and Reasoning*, pages 75–86. Morgan Kaufmann, 1992. Cited on page **136**.

Janusz A. Brzozowski. Derivatives of regular expressions. *Journal of the ACM*, pages 481–494, 1964. Cited on page **44**.

Kanianthra M. Chandy and Jayadev Misra. How processes learn. In *Proceedings of the Fourth Annual ACM Symposium on Principles of Distributed Computing*, pages 204–214, 1985. Cited on page **60**.

Mika Cohen and Mads Dam. A complete axiomatization of knowledge and cryptography. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science*, pages 77–88, 2007. Cited on page **45**.

John H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall Mathematics Series. Chapman and Hall, 1971. Cited on page **44**.

Francien Dechesne and Yanjing Wang. Dynamic epistemic verification of security protocols: Framework and case study. In *Proceedings of the Workshop on Logic, Rationality and Interaction*, Texts in Computer Science, pages 129–144, 2007. Cited on page **40**.

Paul-Olivier Dehaye, Daniel Ford, and Henry Segerman. One hundred prisoners and a light bulb. *Mathematical Intelligencer*, 24(4):53–61, 2003. Cited on page **56**.

Ronald Fagin and Joseph Y. Halpern. Belief, awareness and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988. Cited on page **37**.

Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. The MIT Press, 1995. Cited on pages **37**, **40**, and **60**.

Jelle Gerbrandy. The surprise examination in dynamic epistemic logic. *Synthese*, 155:21–33, 2007. Cited on page **160**.

Jelle Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, ILLC, Amsterdam, 1999. Cited on pages **158** and **160**.

Jelle Gerbrandy and Willem Groeneveld. Reasoning about information change. *Journal of Logic, Language and Information*, 6(2):147–169, 1997. Cited on page **40**.

Edmund Gettier. Is justified true belief knowledge? *Analysis*, 23:121–123, 1963. Cited on page **153**.

Robert Goldblatt. *Logics of Time and Computation, Second Edition, Revised and Expanded*. CSLI Lecture Notes. The University of Chicago Press, 1992. First edition 1987. Cited on page **140**.

Jim Gray. Notes on data base operating systems. In *Operating Systems, An Advanced Course*, pages 393–481. Springer-Verlag, 1978. Cited on page **2**.

Adam Grove. Two modellings for theory change. *Journal of Philosophical Logic*, 17:157–170, 1988. Cited on page **140**.

Joseph Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, 1990. Cited on pages **50** and **60**.

David Harel. Dynamic logic. In Dov Gabbay and Franz Guenther, editors, *Handbook of Philosophical Logic*, pages 497–604. Reidel, 1984. Volume II. Cited on page **138**.

Tomohiro Hoshi. *Epistemic Dynamics and Protocol Information*. PhD thesis, Stanford University, 2009. Cited on page **40**.

Tomohiro Hoshi and Audrey Yap. Dynamic epistemic logic with branching temporal structures. *Synthese*, 169(2):259–281, 2009. Cited on page **40**.

Cor A. J. Hurkens. Spreading gossip efficiently. *Nieuw Archief voor Wiskunde*, 5/1(2):208–210, 2000. Cited on pages **39** and **54**.

Barteld Kooi. Expressivity and completeness for public updates via reduction axioms. *Journal of Applied Non-Classical Logics*, 16(2), 2007. Cited on page **160**.

Dexter Kozen and Rohit Parikh. An elementary proof of the completeness of PDL. *Theoretical Computer Science*, 14:113–118, 1981. Cited on pages **9** and **137**.

Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978. Cited on pages **60** and **65**.

Christian List and Philip Pettit. On the many as one. *Philosophy and Public Affairs*, 33(4):377–390, 2005. Cited on pages **7**, **133**, and **147**.

Peter Montague. A new disinformation campaign. New York Times, April 29, 1998. Cited on page **154**.

Abhaya C. Nayak. Iterated belief change based on epistemic entrenchment. *Erkenntnis*, 41:353–390, 1994. Cited on page **144**.

Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. MIT Press, 1992. Cited on page **165**.

Eric Pacuit. Logics of informational attitudes and informative actions. *Journal of the Council of Indian Philosophy*, 27(2), 2010. Cited on page **60**.

Eric Pacuit and Rohit Parikh. Reasoning about communication graphs. In Johan van Benthem, Benedikt Löwe, and Dov Gabbay, editors, *Interactive Logic*, volume 1 of *Texts in Logic and Games*. Amsterdam University Press, 2007. Cited on pages **40**, **54**, **56**, **60**, and **66**.

Rohit Parikh and Ram Ramanujam. Distributed processes and the logic of knowledge. In *Proceedings of the Conference on Logic of Programs*, pages 256–268, 1985. Cited on page **40**.

Rohit Parikh and Ram Ramanujam. A knowledge based semantics of messages. *Journal of Logic, Language and Information*, 12(4):453–467, 2003. Cited on pages **40**, **60**, and **66**.

Marc Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12:149–166, 2002. Cited on page **133**.

Jan A. Plaza. Logics of public communications. In *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989. Cited on page **157**.

Gordon Plotkin. An operational semantics for CSP. In *Formal Description of Programming Concepts II*, pages 199–225. North Holland, 1983. Cited on page **82**.

Floris Roelofsen. Exploring logical perspectives on distributed information and its dynamics. Master's thesis, University of Amsterdam, 2005. Cited on page **40**.

Chiaki Sakama, Martin Caminada, and Andreas Herzig. A logical account of lying. In *JELIA 2010*, volume 6341 of *Lecture Notes in Computer Science*, pages 286–299, 2010. Cited on page **153**.

Krister Segerberg. A completeness theorem in the modal logic of programs. *Universal Algebra*, 9:31–46, 1982. Cited on page **137**.

Nikolay V. Shilov and Natalya O. Garanina. Model checking knowledge and fixpoints. In *Fixed Points in Computer Science*, pages 25–39, 2002. Cited on page **46**.

Floor Sietsma. Model checking for dynamic epistemic logic with factual change. Bachelor's thesis, University of Amsterdam, 2007. Cited on pages **167** and **171**.

Floor Sietsma and Krzysztof R. Apt. Common knowledge in email exchanges. *ACM Transactions on Computational Logic*, 2012. To appear. Cited on page **6**.

Floor Sietsma and Jan van Eijck. Multi-agent belief revision with linked plausibilities. In *Logic and the Foundations of Game and Decision Theory - LOFT 8*, pages 174–189, 2008. Cited on page **7**.

Floor Sietsma and Jan van Eijck. Message passing in a dynamic epistemic logic setting. In *Proceedings of the Thirteenth Conference on Theoretical Aspects of Rationality and Knowledge*, pages 212–220, 2011. Cited on page **5**.

Floor Sietsma and Jan van Eijck. Action emulation between canonical models. In *Proceedings of the 10th Conference on Logic and the Foundations of Game and Decision Theory*, 2012. Cited on page **6**.

St. Augustine. De mendacio. In P. Schaff, editor, *A Select Library of the Nicene and Post-Nicene Fathers of the Christian Church*, volume 3 (1956). Eerdmans, 1988. URL `http://www.newadvent.org/fathers/`. Translated by Rev. H. Browne. Cited on pages **151** and **152**.

Edward Szpilrajn. Sur l'extension de l'ordre partiel. *Fundamenta Mathematicae*, 16:386–389, 1930. Cited on page **83**.

Alan D. Taylor. *Social Choice and the Mathematics of Manipulation*. Cambridge University Press, 2005. Cited on page **133**.

Johan van Benthem. 'One is a lonely number': On the logic of communication. In *Logic Colloquium '02*, pages 96–129. ASL & A.K. Peters, 2002. Cited on page **40**.

Johan van Benthem. Dynamic logic for belief revision. *Journal of Applied Non-Classical Logics*, 2:129–155, 2007. Cited on page **144**.

Johan van Benthem and Fenrong Liu. Dynamic logic and preference upgrade. *Journal of Applied Non-Classical Logics*, 14(2):157–182, 2004. Cited on page **144**.

Johan van Benthem, Jan van Eijck, and Barteld Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006. Cited on pages **24**, **34**, **37**, **60**, **133**, **142**, **143**, **155**, **159**, **160**, and **170**.

Johan van Benthem, Jelle Gerbrandy, Tomohiro Hoshi, and Eric Pacuit. Merging frameworks for interaction. *Journal of Philosophical Logic*, 38(5):491–526, 2009a. Cited on pages **40** and **50**.

Johan van Benthem, Jelle Gerbrandy, and Barteld Kooi. Dynamic update with probabilities. *Studia Logica*, 93:67–96, 2009b. Cited on page **168**.

Ron van der Meyden and Nikolay V. Shilov. Model checking knowledge and time in systems with perfect recall. In *Proceedings of the 19th Conference on the Foundations of Software Technology and Theoretical Computer Science*, volume 1738 of *Lecture Notes in Computer Science*, 1999. Cited on page **46**.

Hans van Ditmarsch. *Knowledge Games*. PhD thesis, Groningen University, 2000. Cited on page **40**.

Hans van Ditmarsch. Comments on 'the logic of conditional doxastic actions'. In *New Perspectives on Games and Interaction*, volume 4 of *Texts in Logic and Games*, pages 33–44. Amsterdam University Press, 2008. Cited on page **160**.

Hans van Ditmarsch and Tim French. Becoming aware of propositional variables. In *Logic and its Applications*, volume 6521 of *Lecture Notes in Computer Science*, pages 204–218. Springer, Berlin/Heidelberg, 2011. Cited on page **37**.

Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library*. Springer, 2006. Cited on page **37**.

Hans van Ditmarsch, Jan van Eijck, Floor Sietsma, and Yanjing Wang. On the logic of lying. In Jan van Eijck and Rineke Verbrugge, editors, *Games, Actions and Social Software*, volume 7010 of *Lecture Notes in Computer Science*, pages 41–72. Springer, 2012. Cited on pages **7** and **160**.

Jan van Eijck. DEMO - A demo of epistemic modelling. In *Interactive Logic - Proceedings of the 7th Augustus de Morgan Workshop*, volume 1 of *Texts in Logic and Games*, pages 305–363, 2007. Cited on page **167**.

Jan van Eijck. Yet more modal logics of preference change and belief revision. In Krzysztof R. Apt and Robert van Rooij, editors, *New Perspectives on Games and Interaction*, volume 4 of *Texts in Logic and Games*, pages 81–104. Amsterdam University Press, 2008. Cited on pages **144** and **146**.

Jan van Eijck and Yanjing Wang. Propositional dynamic logic as a logic of belief revision. In *Proceedings of Wollic '08*, number 5110 in Lecture Notes in Artificial Intelligence, pages 136–148, 2008. Cited on pages **133** and **143**.

Jan van Eijck, Yanjing Wang, and Floor Sietsma. Composing models. *Journal of Applied Non-Classical Logics*, 21:397–425, 2011. Cited on pages **23** and **37**.

Jan van Eijck, Ji Ruan, and Tomasz Sadzik. Action emulation. *Synthese*, 185(1): 131–151, 2012. Cited on pages **113**, **116**, **118**, **119**, **120**, **123**, **127**, **129**, and **131**.

Yanjing Wang. *Epistemic Modelling and Protocol Dynamics*. PhD thesis, ILLC, Amsterdam, 2010. Cited on page **168**.

Yanjing Wang, Lakshmanan Kuppusamy, and Jan van Eijck. Verifying epistemic protocols under common knowledge. In *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 257–266, 2009. Cited on page **40**.

Yanjing Wang, Floor Sietsma, and Jan van Eijck. Logic of information flow on communication channels. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 1447–1448, 2010. Cited on page **5**.

Abstract

The goal of this dissertation is to give a logical representation of the knowledge dynamics that takes place during communication. I present a number of different logical frameworks for a number of different scenarios, ranging from an email conversation where all information that is sent is considered to be true, to a game of Liar's Dice where lying is expected of the players.

In Chapter 3, I present a framework for modeling the knowledge of agents who exchange messages, using Dynamic Epistemic Logic. This framework uses Kripke models to represent the agents' knowledge in a static situation, and action models to update these Kripke models when the situation changes. Because the models are supposed to be finite, and all messages are represented explicitly in the model, the messages that are considered possible by the agents are limited to a finite set. This framework is useful in a situation in which there is a number of rounds in each of which a finite set of new messages becomes available to the agents. These messages can gradually be added to the model.

The framework presented in Chapter 4 is of a more general nature. It models a setting where agents communicate with messages over a specific network in accordance to a certain protocol. This framework is very flexible because the nature of communicative events and the observational power of the agents can be adapted to the situation at hand. It combines properties of the Dynamic Epistemic Logic approach with the perspective of Interpreted Systems.

In Chapter 5 and 6 I focus on email communication specifically. I first study the existence of common knowledge in a group of agents who communicate via emails. Unlike the approach presented in Chapter 3, all possible emails are represented in the model, which is therefore of infinite size. I prove a number of properties of finite states in this infinite model, and show that common knowledge of an email with BCC recipients is rare.

Apart from common knowledge, I consider two new kinds of knowledge: potential and definitive knowledge. These two types of knowledge make a distinction between the assumption that every agent immediately reads every email he re-

ceives, or that every agent has only read the emails he replied to or forwarded. I also present a method to do model checking, even though the model is of infinite size.

Chapter 7 is a study of the properties of action models, which are used to model communicative events. I define a notion of action emulation that signifies when two canonical action models are equivalent. Because every action model has an equivalent canonical action model which can be computed, this gives a general method to determine action model equivalence.

In Chapter 8 I move from knowledge to belief. I use the same Kripke models as for knowledge, only without the assumption that all relations are equivalence relations. I propose a different assumption, namely that the relations are linked. I also give a number of updates of these models that preserve this property, representing communicative events.

Finally, Chapter 9 gives different perspectives on the issue of lying. It includes a complete logic of manipulative updating, which can be used to represent the effects of lying in a group of agents. I also analyze a game of Liar's Dice and implement this scenario in the model checker DEMO. Furthermore, I show that in a game where lying is considered normal, a lie is no longer a lie: because the agents who hear the lie do not believe it, no false belief is created.

Samenvatting

Het doel van dit proefschrift is het geven van een logische representatie van de kennisdynamica die plaatsvindt tijdens communicatie. Ik presenteer een aantal verschillende logische systemen voor verschillende scenario's, variërend van een email conversatie waarin alle verzonden informatie als waar wordt beschouwd, tot een spelletje blufpoker waarbij liegen van de spelers verwacht wordt.

In Hoofdstuk 3 presenteer ik een systeem voor het modelleren van de kennis van agenten die berichten uitwisselen, waarbij ik gebruik maak van Dynamische Epistemische Logica. Dit systeem gebruikt Kripke modellen om de kennis van de agenten in een statische situatie te representeren, en actiemodellen om deze Kripke modellen bij te werken als de situatie verandert. Omdat ik aanneem dat de modellen eindig zijn, en omdat alle berichten expliciet worden gerepresenteerd in het model, zijn de berichten die de agenten mogelijk achten gelimiteerd tot een eindige verzameling. Dit systeem is nuttig in situaties waarin sprake is van een aantal rondes waarin telkens een eindige verzameling nieuwe berichten voor de agenten beschikbaar wordt. Deze berichten kunnen gradueel worden toegevoegd aan het model.

Het systeem dat gepresenteerd wordt in Hoofdstuk 4 heeft een meer algemeen karakter. Het modelleert een situatie waarin agenten communiceren over een specifiek netwerk, in overeenstemming met een bepaald protocol. Dit systeem is erg flexibel omdat de aard van de communicatieve gebeurtenissen en de observerende vermogens van de agenten kunnen worden aangepast aan de situatie. Het combineert eigenschappen van Dynamische Epistemische Logica met het perspectief van Geïnterpreteerde Systemen.

In Hoofdstuk 5 en 6 concentreer ik me op email communicatie. Ik bestudeer eerst het ontstaan van gezamenlijke kennis in een groep agenten die communiceren via email. In tegenstelling tot de aanpak van Hoofdstuk 3 worden in dit model alle mogelijke emails gerepresenteerd in het model, wat dan ook van oneindige grootte is. Ik bewijs een aantal eigenschappen van de eindige toestanden binnen dit model, en ik laat zien dat gezamenlijke kennis van een email met BCC ontvangers

erg zeldzaam is.

Buiten gezamenlijke kennis beschouw ik twee nieuwe vormen van kennis: potentiële en definitieve kennis. Deze twee vormen van kennis maken een onderscheid tussen de aanname dat iedere agent iedere email die hij ontvangt onmiddellijk leest, en de aanname dat iedere agent alleen de emails heeft gelezen die hij heeft beantwoord of doorgestuurd. Ik presenteer ook een manier om de waarheid van een formule in mijn model te controleren, ondanks het feit dat het model oneindig groot is.

Hoofdstuk 7 is een studie van de eigenschappen van actiemodellen, die gebruikt worden om communicatieve gebeurtenissen te modelleren. Ik definieer een notie van actie emulatie die aangeeft wanneer twee canonieke actiemodellen equivalent zijn. Omdat ieder actiemodel een equivalent canoniek actiemodel heeft dat ook berekend kan worden, geeft dit een algemene methode om te beslissen of twee actiemodellen equivalent zijn.

In Hoofdstuk 8 verschuift mijn aandacht van kennis naar geloof. Ik gebruik dezelfde Kripke modellen als voor kennis, alleen zonder de aanname dat alle relaties equivalentierelaties zijn. Ik stel een nieuwe eis voor, namelijk dat de relaties verbonden zijn. Ik geef ook een aantal manieren om deze modellen bij te werken die deze eis respecteren, en communicatieve gebeurtenissen kunnen representeren.

Als laatste geeft Hoofdstuk 9 verschillende perspectieven op het concept van liegen. Ik geef onder andere een complete logica van manipulatieve communicaties, die gebruikt kan worden om de effecten van liegen in een groep agenten te representeren. Ik analyseer ook een spelletje blufpoker en ik implementeer dit scenario in de modelbevrager DEMO. Ik laat zien dat in een spel waarin het normaal is om te liegen, een leugen niet langer een leugen is: omdat de agenten die de leugen horen hem niet geloven, wordt er geen onwaar geloof gecreëerd.

Titles in the ILLC Dissertation Series:

ILLC DS-2006-01: **Troy Lee**
Kolmogorov complexity and formula size lower bounds

ILLC DS-2006-02: **Nick Bezhanishvili**
Lattices of intermediate and cylindric modal logics

ILLC DS-2006-03: **Clemens Kupke**
Finitary coalgebraic logics

ILLC DS-2006-04: **Robert Špalek**
Quantum Algorithms, Lower Bounds, and Time-Space Tradeoffs

ILLC DS-2006-05: **Aline Honingh**
The Origin and Well-Formedness of Tonal Pitch Structures

ILLC DS-2006-06: **Merlijn Sevenster**
Branches of imperfect information: logic, games, and computation

ILLC DS-2006-07: **Marie Nilsenova**
Rises and Falls. Studies in the Semantics and Pragmatics of Intonation

ILLC DS-2006-08: **Darko Sarenac**
Products of Topological Modal Logics

ILLC DS-2007-01: **Rudi Cilibrasi**
Statistical Inference Through Data Compression

ILLC DS-2007-02: **Neta Spiro**
What contributes to the perception of musical phrases in western classical music?

ILLC DS-2007-03: **Darrin Hindsill**
It's a Process and an Event: Perspectives in Event Semantics

ILLC DS-2007-04: **Katrin Schulz**
Minimal Models in Semantics and Pragmatics: Free Choice, Exhaustivity, and Conditionals

ILLC DS-2007-05: **Yoav Seginer**
Learning Syntactic Structure

ILLC DS-2008-01: **Stephanie Wehner**
Cryptography in a Quantum World

ILLC DS-2008-02: **Fenrong Liu**
Changing for the Better: Preference Dynamics and Agent Diversity

ILLC DS-2008-03: **Olivier Roy**
Thinking before Acting: Intentions, Logic, Rational Choice

ILLC DS-2008-04: **Patrick Girard**
Modal Logic for Belief and Preference Change

ILLC DS-2008-05: **Erik Rietveld**
Unreflective Action: A Philosophical Contribution to Integrative Neuroscience

ILLC DS-2008-06: **Falk Unger**
Noise in Quantum and Classical Computation and Non-locality

ILLC DS-2008-07: **Steven de Rooij**
Minimum Description Length Model Selection: Problems and Extensions

ILLC DS-2008-08: **Fabrice Nauze**
Modality in Typological Perspective

ILLC DS-2008-09: **Floris Roelofsen**
Anaphora Resolved

ILLC DS-2008-10: **Marian Counihan**
Looking for logic in all the wrong places: an investigation of language, literacy and logic in reasoning

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder…": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems

ILLC DS-2010-04: **Daisuke Ikegami**
Games in Set Theory and Logic

ILLC DS-2010-05: **Jarmo Kontinen**
Coherence and Complexity in Fragments of Dependence Logic

ILLC DS-2010-06: **Yanjing Wang**
Epistemic Modelling and Protocol Dynamics

ILLC DS-2010-07: **Marc Staudacher**
Use theories of meaning between conventions and social norms

ILLC DS-2010-08: **Amélie Gheerbrant**
Fixed-Point Logics on Trees

ILLC DS-2010-09: **Gaëlle Fontaine**
Modal Fixpoint Logic: Some Model Theoretic Questions

ILLC DS-2010-10: **Jacob Vosmaer**
Logic, Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.

ILLC DS-2010-11: **Nina Gierasimczuk**
Knowing One's Limits. Logical Analysis of Inductive Inference

ILLC DS-2010-12: **Martin Mose Bentzen**
Stit, Iit, and Deontic Logic for Action Types

ILLC DS-2011-01: **Wouter M. Koolen**
Combining Strategies Efficiently: High-Quality Decisions from Conflicting Advice

ILLC DS-2011-02: **Fernando Raymundo Velazquez-Quesada**
Small steps in dynamics of information

ILLC DS-2011-03: **Marijn Koolen**
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

ILLC DS-2011-04: **Junte Zhang**
System Evaluation of Archival Description and Access

ILLC DS-2011-05: **Lauri Keskinen**
Characterizing All Models in Infinite Cardinalities

ILLC DS-2011-06: **Rianne Kaptein**
Effective Focused Retrieval by Exploiting Query Context and Document Structure

ILLC DS-2011-07: **Jop Briët**
Grothendieck Inequalities, Nonlocal Games and Optimization

ILLC DS-2011-08: **Stefan Minica**
Dynamic Logic of Questions

ILLC DS-2011-09: **Raul Andres Leal**
Modalities Through the Looking Glass: A study on coalgebraic modal logic and their applications

ILLC DS-2011-10: **Lena Kurzen**
Complexity in Interaction

ILLC DS-2011-11: **Gideon Borensztajn**
The neural basis of structure in language

ILLC DS-2012-01: **Federico Sangati**
Decomposing and Regenerating Syntactic Trees

ILLC DS-2012-02: **Markos Mylonakis**
Learning the Latent Structure of Translation

ILLC DS-2012-03: **Edgar José Andrade Lotero**
Models of Language: Towards a practice-based account of information in natural language

ILLC DS-2012-04: **Yurii Khomskii**
Regularity Properties and Definability in the Real Number Continuum: idealized forcing, polarized partitions, Hausdorff gaps and mad families in the projective hierarchy.

ILLC DS-2012-05: **David García Soriano**
Query-Efficient Computation in Property Testing and Learning Theory

ILLC DS-2012-06: **Dimitris Gakis**
Contextual Metaphilosophy - The Case of Wittgenstein

ILLC DS-2012-07: **Pietro Galliani**
The Dynamics of Imperfect Information

ILLC DS-2012-08: **Umberto Grandi**
Binary Aggregation with Integrity Constraints

ILLC DS-2012-09: **Wesley Halcrow Holliday**
Knowing What Follows: Epistemic Closure and Epistemic Logic

ILLC DS-2012-10: **Jeremy Meyers**
Locations, Bodies, and Sets: A model theoretic investigation into nominalistic mereologies

ILLC DS-2012-11: **Floor Sietsma**
Logics of Communication and Knowledge