NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

122

This reproduction is the best copy available.

UMI®

TOPICS IN SOCIAL SOFTWARE

INFORMATION IN STRATEGIC SITUATIONS

by

ERIC PACUIT

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

2005

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

4-18-05 (Rohit Parikh)
Date — Chair of Examining Committee

4-18-05 (Ted Brown)
Date — Executive Officer

Horacio Arló-Costa

Sergei Artemov

Steven Brams

Melvin Fitting

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

TOPICS IN SOCIAL SOFTWARE: INFORMATION IN STRATEGIC SITUATIONS

by

Eric Pacuit

Adviser: Professor Rohit Parikh

Social software is an emerging interdisciplinary field devoted to the design and analysis of social procedures. This new field has recently gained the attention of a wide range of research communities, including computer scientists, game theorists and philosophers. The main idea behind social software is that constructing and verifying social procedures should be pursued as systematically as computer software is pursued by computer scientists. The logical systems developed in this thesis are intended to facilitate such an analysis.

Although the analogy between computer software and social software is strong, there are some important differences. For example, two issues which are important for an analysis of social procedures but less crucial for computer software are the exchange of (and occasional hiding of) information, and the provision of incentives. Concurrency theory, cryptography and distributed computing have all addressed the first issue. However, many of the underlying assumptions in these fields make applying these results to social procedures unrealistic. The second issue has been more or less the province of game theory. But game theory tends to study the area in rather simple terms lacking the

sophisticated tools of computer science such as modularization or data types. The objective of this thesis is to develop formal frameworks which may be used to verify social procedures, with special attention paid to naturally modeling the flow of information in a social situation.

to Lauren, Laura, Mom and Dad

Acknowledgments

It was a pleasure and an honor to work with my adviser, Rohit Parikh. I will always be grateful for our intellectually stimulating discussions on topics that range from politics to the foundations of mathematics to Indian movies (and, time permitting, our research). I would also like to thank the rest of my dissertation committee for their insightful comments and discussions: Horacio Arló-Costa, Sergei Artemov, Steven Brams and Melvin Fitting.

I thank the Knowledge, Games and Beliefs Group of CUNY for allowing me to present much of this work at various stages of development. In particular, I thank the following friends and colleagues for discussions which have all had an impact on this thesis: Samir Chopra, Eva Cogan, Walter Dean, Kevin Fenwick, Rave Harpaz, Roman Kuznets, Bryan Renee, Samer Salame, Chris Steinsvold and Noson Yanofsky. I also thank Johan van Benthem, Hans van Ditmarsch, Barteld Kooi and Marc Pauly for discussions abroad which have influenced my writing.

I thank Brooklyn College and the Research Foundation of CUNY for their financial support. I would also like to thank Joe Driscoll for not actually charging me for using the copy machine.

Of course none of this would be possible without the support of my friends and family. Thanks goes to Dick and Pat Jay whose generosity always amazes me. A special thanks goes to my 8th grade math teacher (who happens to be my mother), my junior high

math team coach (who happens to be my father) and their high school math teacher (who happens to be my grandmother). Finally, I thank Lauren for her love and support and for always finding a way to distract me from my work.

Table of Contents

1 Introduction **1**

1.1 What is Social Software? 2

1.1.1 Models of Social Situations 4

1.1.2 A Theory of Correctness of Social Procedures 5

1.1.3 Designing Social Procedures 9

1.2 Social Software for the Game Theorist 9

1.3 Social Software for the Computer Scientist 12

1.4 Overview 13

2 History Based Structures **16**

2.1 History Based Structures 19

2.2 Logics of Knowledge and Time 22

2.2.1 Histories or Runs? 33

2.3 Extensive Games 39

3 Knowledge Based Obligation **43**

3.1 Introduction 44

3.2 Actions 49

3.3 Values 52

3.3.1 Comparison with Horty 55

3.4 Default Histories 57

3.4.1 Default Obligations 60

3.5 Putting Everything Together 63

3.6 Formalizing the Examples 68

3.6.1 Common Knowledge of Ethicality 77

3.7 Programming the Agents 79

3.8 Conclusions 82

4 Communication Graphs **85**

4.1 From Topologic to Communication Graphs 87

4.2 The Logic of Communication Graphs 93

4.2.1 Semantics 94

4.2.2 Surface Knowledge 100

4.2.3 Axioms and Decidability 102

4.2.4 Connection with Communication Graphs 108

4.3 Conclusions and Further Work 108

5 Strategic Voting **113**

5.1 Introduction 114

5.2 A Formal Voting Model 117

5.3 Conclusion and Further Work 133

6 Conclusions **135**

Bibliography **143**

Chapter 1

Introduction

Donald Knuth begins his remarkable series of books, *The Art of Computer Programming*, with the statement "The notion of an algorithm is basic to all computer programming, and so we should begin with a careful analysis of this concept." In other words, in order to understand the complex behavior of a computer, one needs a systematic and rigorous analysis of algorithms, or procedures. The key idea behind *social software* is to apply this simple idea to social situations. That is, a systematic and rigorous analysis of *social procedures* can help us understand social interactions and may lead to a more "efficient" society. This idea was put forward by Rohit Parikh [74], and has recently gained the attention of a wide range of research communities, including computer scientists, game theorists and philosophers. The objective of this thesis is to develop logical systems that can be used to reason about multi-agent interactive situations relevant for the analysis

of social procedures.

1.1 What is Social Software?

Starting with [74] and more recently in [76, 77, 75], Parikh defines social software by way of various illustrative examples. Essentially, there are two ways in which a procedure can fit into the social software paradigm. First of all, a procedure may be truly social in that several agents are required even in the execution of the procedure. Standard examples are voting procedures, such as plurality voting or approval voting, or fair division algorithms, such as adjusted winner or the many cake-cutting algorithms. Secondly, even if a procedure does not *require* a group for its execution, it may still fit into the social software paradigm. These are procedures set up by society and intended to be performed by single agents within the context of a group of agents. Examples include procedures that universities set up that students must follow in order to drop a class or the procedures hospitals set up to ensure the necessary flow of information from a patient to a doctor.

From the point of view of someone designing a social procedure, as soon as beliefs and utilities can be attributed to the agent(s) executing the procedure, the procedure should be thought of as social software. After all, when designing computer software, programmers do not worry that the computer may suddenly not "feel like" performing the next step of the algorithm. But in a setting where agents have individual preferences,

such considerations must be taken into account. In fact, this suggests a third way in which procedures can be analyzed within the social software paradigm - individual agents executing procedures in isolation. For example, an agent following a recipe in order to make peanut-butter chocolate chip cookies. However, from this point of view, certain philosophical questions about the nature of procedures, or algorithms, and human knowledge and beliefs become much more important. In this thesis, the fact that a group of agents is somehow involved in the execution of a social procedure will play an essential role. As a consequence, the study of social software draws on results not only from artificial intelligence, distributed computing and philosophical logic but also from game theory, especially mechanism design, and economics.

So far we have only explained the type of situations we have in mind and have not yet provided an adequate *definition* of social software. We will now attempt to rectify this situation. Social software is an interdisciplinary research program that combines mathematical tools and techniques from game theory and computer science in order to analyze and design social procedures. Research in social software can be divided into three different but related categories: modeling social situations, developing a theory of correctness of social procedures and designing social procedures. The next three subsections will describe these areas in more detail.

1.1.1 Models of Social Situations

One of the central issues in social software and the focus of this thesis is the development of appropriate models of multi-agent interactive situations. If one wants a careful and rigorous analysis of social procedures, one needs to begin with a realistic model of multi-agent interaction. The search for such models has occupied researchers in a number of different disciplines including (but not limited to), game theory, philosophy, artificial intelligence and distributed computing. What is needed from the social software point of view are formal models in which our intuitions about social procedures can be refined and tested.

It is important to be clear about exactly what is being proposed. Perhaps it is too much to ask for a general theory which explains all social interactions, i.e., a "theory of everything" for the social sciences. If at all possible, such a theory would require collaboration among a vast array of research communities including psychologists, biologists, cognitive scientists and so on. What *is* being developed is a collection of logical systems intended to be used to formalize multi-agent interactive situations relevant for the analysis of social procedures. These frameworks are developed from different points of view and are governed by different assumptions about the agents involved.

There are many ways one could formalize social situations in which the agents are assumed to be executing some procedure. In this thesis, the role of information and its dynamics in strategic multi-agent situations is a central issue. To be more precise, we

focus on the following two issues:

1. **Knowledge, Actions and Obligations:** It is natural to assume that the agents' choices, at least partially, depend on their current states of information. This is particularly important when reasoning about what agents ought to do. Certainly an agent cannot be faulted for not performing an action whose need it did not know about. How should this dependency of actions on knowledge be modeled?

2. **Updating Information:** Just as registers in a computer are continually updated as a program is executed, the information of each agent changes as a social procedure is executed. How should information update be represented? How does this affect the models of knowledge and belief?

1.1.2 A Theory of Correctness of Social Procedures

Just as one can prove that a certain implementation of a sorting algorithm is correct, perhaps one can prove that a certain piece of social software is correct. In computer science, it is often convenient to view a computational procedure, or a program, as a relation on a set of states, where a state can be thought of as a function that assigns a value to every possible variable and a truth value to all propositions. This approach was first proposed by Pratt ([90]) and is based on the work of Floyd ([39]) and Hoare ([59]). Harel, Kozen and Tiuryn ([53]) provide a very thorough discussion of computational

procedures from this point of view. The first step towards a formal logic of social procedures influenced by this analysis of computational procedures is Parikh's game logic ([72, 73]). Game logic is intended to formalize situations where agents have directly opposing preferences. Marc Pauly took the analysis one step further in his dissertation, *The Logic of Social Software* [85], in which he developed a logic for reasoning about coalitions [86, 87]. Recently, Pauly has developed a formal framework in the style of Hoare [59] for proving correctness of social procedures [84].

However, none of the frameworks discussed above provide an explicit representation of agents' knowledge. Finding an appropriate representation is an important step towards a formal theory of correctness of social procedures. As discussed above, one of the main aspects of social procedures is that a group of agents is somehow involved. In this multi-agent setting it is natural to assume that the agents do not all have exactly the same information. And so issues about how each agent represents the other agents' information becomes important. In particular, notions such as common knowledge, distributed knowledge, and other levels of knowledge[1] are very relevant when trying to formalize correctness conditions of many social procedures.

To illustrate the above point, consider the fair division algorithm *adjusted winner* [14]. Adjusted winner is an algorithm to fairly distribute divisible goods among two people. The procedure is discussed in detail in [14]. Instead of providing a formal

[1]See [79, 78] for more on the application of levels of knowledge *below* common knowledge to game theoretic situations.

description of the procedure, we will look at an illustrative example. More theoretical and practical information can be found in the books [14, 15]. Suppose Ann and Bob are dividing three goods: A, B, and C. Adjusted winner begins by giving both Ann and Bob 100 points to divide among the three goods. Suppose that Ann and Bob assign points according to the following table.

Item	Ann	Bob
A	<u>10</u>	5
B	<u>65</u>	45
C	25	<u>50</u>
Total	100	100

The second step of the procedure is to give A and B to Ann since she assigned more points to those items and item C to Bob. However this is not an equitable outcome since Ann has received 75 points while Bob only received 50 points. We must transfer some of Ann's goods to Bob. However, even giving all of item A to Bob will not create an equitable division since Ann now has 65 points, but Bob has only 60 points. In order to create equitability, we must transfer part of item B from Ann to Bob. Let p be the proportion of item B that Ann will keep. Then

$$65p = 100 - 45p$$

yielding $p = 100/110 = 0.9090$, so Ann will keep 91% of item B and Bob will get

9% of item B. Thus both Ann and Bob receive 59.09 points. This allocation (Ann receives 91% of item B and Bob receives all of item A and item C plus 9% of item B) is *envy-free*, *equitable* and *efficient*. See [14] for a formal definition of these properties and proofs that the adjusted winner procedure satisfies these properties. What we are interested in is whether Ann can improve her total allocation by misrepresenting her preferences. It turns out that she can improve her allocation. If Ann announces that her allocation is 6 points for A, 55 points for B and 39 points for C, then the adjusted winner algorithm will give all of B to Ann and all of A and C to Bob. This results in Ann receiving a total of 65 points (according to her original preferences) while Bob receives 55 points. Now this type of deception is only possible in the extreme case that Ann knows Bob's preferences, but *Bob does not know Ann's preferences.* Indeed, Brams and Taylor successfully argue in [14] that unless an agent is certain that its preference is private and that its information about the other agent's preference is correct, then that agent is better off reporting its true preference. The point is that part of showing that an *implementation* of adjusted winner is correct must take into account that the each agent's preference must be kept private, or at least that the agents cannot take advantage of whatever information they have about the other agent's preferences. In other words, given the results from Brams and Taylor that adjusted winner is efficient, envy free and equitable, showing that an implementation of adjusted winner is correct reduces to showing that a particular level of knowledge among the agents is maintained

(i.e., each agent's preference is kept private).

1.1.3 Designing Social Procedures

An important test of any theory is how well it can be applied to real life situations. There are a wealth of social procedures described in the game theory literature. For instance, see [14] for a discussion of a number of different fair division algorithms. Attempts to formalize these procedures create a wide range of interesting problems for logicians. For example, see [67, 93] for a logic that axiomatizes the concept of majority. It will be very interesting to see if logical analyzes can suggest refinements of existing social procedures or help create new social procedures.

1.2 Social Software for the Game Theorist

Starting with von Neumann and Morgenstern, and later with the seminal work of Nash, game theorists have developed elegant mathematical theories that can formalize situations similar to the one described above. Mechanism design (or implementation theory) studies situations in which the agents' preferences are given together with a desired set of outcomes, and asks what interactive situation (a game) can be designed in which the outcomes can be achieved assuming the agents act according to their given preferences. See [65] Chapter 10 for more information. Essentially, a piece of social software is cre-

ated. Pauly and Wooldridge [88] and Halpern [50] independently have recently argued that formal logical systems can be useful in this setting.

Evidence of the usefulness of a rigorous analysis of social procedures can already be seen in many areas of economics and social choice theory. Classic results such as Arrow's Theorem or the Gibbard-Satterthwaite Theorem [45, 95, 96] have had profound effects on social choice theory and voting theory. A more concrete example is the game-theoretic analysis of the procedure used by King Solomon in the well-known biblical story. In this story, King Solomon is faced with two women each claiming that a baby is her own child. Solomon threatens to cut the baby in half causing one of the mothers to rescind her claim of motherhood; thus revealing herself as the true mother. However a formal analysis of this procedure, first pointed out by Glazer and Ma in 1989 [46], reveals a mistake in Solomon's procedure. It is possible for the false mother to outwit Solomon by misrepresenting her actual preference and claim, as the true mother would, that the baby should be given to the other woman. Glazer and Ma suggest a small refinement of the procedure (involving the use of money) creating a "strategy-proof" procedure. What is missing from these analyzes is an explicit and rigorous description of what it means for a social procedure to be correct. An important objective of social software is to fill this gap.

At this point, we have only argued for a formal and rigorous analysis of social procedures and have said nothing about why *logic* should be used for such an analysis.

Presumably, many researchers will agree that rigor is important when analyzing social procedures, but may hesitate to say that a logical analysis is also important. Using logic to analyze social procedures is an important part of social software research. Certainly, it is hard to argue that logic is not an important tool if the goal is to mechanize the analysis of social procedures. But what if the goal is not a computer program, but rather the analysis of a social procedure itself? Can a logical analysis still be of use? Answering this question raises a number of very interesting an important issues which have been addressed by a number of different authors (for example see [70, 98, 99, 6, 13, 100, 5, 92]). A complete answer to this question will be an interesting digression, but a digression nonetheless. Instead we let Bacharach have the final say:

> Game theory is full of deep puzzles, and there is often disagreement about proposed solutions to them. The puzzlement and disagreement are neither empirical nor mathematical but, rather, concern the meanings of fundamental concepts ('solution', 'rational', 'complete information') and the soundness of certain arguments...Logic appears to be an appropriate tool for game theory both because these conceptual obscurities involve notions such as reasoning, knowledge and counter-factuality which are part of the stock-in-trade of logic, and because it is a prime function of logic to establish the validity or invalidity of disputed arguments. — M.O.L. Bacharach [6]

1.3 Social Software for the Computer Scientist

The existence of more and more autonomous programs has prompted the study of *computational* mechanism design which applies economic principles to the design of multi-agent systems. A formal theory of correctness of social procedures can be used to develop computational tools that can verify interactive multi-agent protocols, such as online auctions. See [30, 94, 97] for discussions of the relevant issues.

For example, if I want to fly to India, instead of searching for the best deal and purchasing the ticket myself, I may give a set of constraints to an Internet agent (see [109] for issues related to designing such autonomous computational agents), and then send the agent out to find me the best possible price (given my constraints) and return with the purchased ticket. Of course, at the other end the airlines may also have an autonomous agent designed to sell tickets at a certain price. Now, these agents will interact and bargain according to some predefined procedure, or protocol. How do we know that the procedure the agents follow, which after all is just another piece of code, actually implements the intended social interaction? In other words, the interaction of the internet agents is intended to somehow mimic a real-life interaction between a buyer and a seller. A formal theory of correctness of social software could be used to determine whether the procedure under consideration, in this case the procedure that the seller agent and the buyer agent use to come to a deal, is "sound" and "complete" for its intended interpretation. By "sound", I mean that by following the rules of the

procedure no unintended consequences are generated. For example, the buyer should not be able to leave with a ticket without spending any money. "Completeness" is slightly more complicated. Suppose that P is some procedure intended to mimic some social interaction, say S. Now there are many possible consequences of the social interaction S. These consequences may be described by propositional formulas, such as "the buyer has a ticket to India", but also may be knowledge-theoretic in nature, such as the buyer believes that it got the best deal. The procedure P would be "complete" for situation S if all such consequences could be achieved by the agents.

Finally, we point to another application of social software relevant for both computer scientists and game theorists. The study of rational agents is central to both game theory and artificial intelligence. Simply put, an agent is an entity situated in an environment who is capable of performing actions that somehow change the environment; a *rational agent* chooses actions that are in its own best interests. In [102], van der Hoek and Wooldridge discuss the importance and difficulty of developing a logic of rational agency. The logical systems developed in this thesis deal with many of the same questions as those posed in [102].

1.4 Overview

This thesis uses techniques from modal logic, especially epistemic logic, and game theory to develop formal models appropriate for the analysis of social software. The basics of

modal logic and game theory will be assumed throughout the thesis, see [10] for more information on modal logic and [65] for more information on game theory.

The basic formal framework used is a history based model. The idea is that each agent has a set of possible actions, or choices, and at each moment some event takes place. In computer science, history based models have been used to model computations in a distributed environment. See [34] for a thorough discussion. In the game theory literature, the history based models are called extensive games. The reader is referred to the textbook [65] for a discussion of extensive games. Chapter 2 provides an introduction to this framework.

Chapter 3 reports on joint work with Rohit Parikh and Eva Cogan. Starting with the intuition that agents cannot be expected to perform actions they are unaware of, a multi-agent logic of knowledge, action and obligation is developed. Various deontic dilemmas are studied that illustrate the dependency of an agent's obligation on its knowledge.

Chapter 4 is the result of joint work with Rohit Parikh. In [66], agents are assumed to have some private information at the outset, but may refine their information by acquiring information possessed by other agents, possibly via yet other agents. A multi-agent modal logic with knowledge modalities and a modality representing communication among agents is introduced and shown to be decidable.

Chapter 5 looks at voting theory from the social software point of view. This chapter is based on joint work with Samir Chopra and Rohit Parikh [23].

Finally, in Chapter 6 we conclude and point to further directions of research.

Chapter 2

History Based Structures

This chapter takes up the challenge discussed in Section 1.1.1 — a mathematical model appropriate for analyzing social software is developed. Up to now, what is meant by "appropriate for analyzing social software" has been left rather vague. In fact, in this thesis, no attempt will be made to formalize this notion. Instead, we will argue that the basic framework developed in this chapter can be extended to represent many aspects of social situations relevant for the analysis of social procedures. In particular, we will demonstrate that this framework can be naturally extended to handle two of the main points discussed in the introduction:

1. An agent's choice of action (especially obligatory action) depends on the agent's state of knowledge. (Chapter 3)

2. The occurrence of an event, such as communication, changes the agents' states of

knowledge. (Chapter 4)

Suppose we fix a social interactive situation involving a (finite) set of agents $\mathcal{A}$. What aspects are relevant for the analysis of social procedures? First of all, since the intended application of our models is to study agents *executing a procedure*, it is natural to assume the existence of a global discrete clock (whether the agents have access to this clock is another issue that will be discussed shortly). The natural numbers $\mathbb{N}$ will be used to denote clock ticks. Note that this implies that we are assuming a finite past with a possibly infinite future. The basic ideas is that at each clock tick, or moment, some *event* takes place. This leads us to our second basic assumption.

Typically, no agent will have *all* the information about a situation. For one thing agents are computationally limited and can only process a bounded amount of information. Thus if a social situation can only be described using more bits of information than an agent can process, then that agent can only maintain a portion of the total information describing the situation. Also, the observational power of an agent is limited. For example, suppose that the exact size of a piece of wood is the only relevant piece of information about some situation. While an agent may have enough memory to remember this single piece of information, measuring devices are subject to error. Furthermore, some agents may not *see*, or be aware of, many of the events that take place. Therefore it is fair to assume that two different agents may have different views, or interpretations, of the same situation.

Starting with von Wright's work in the 1950s and Hintikka's seminal book, *Knowledge and Beliefs* [58], there has been a lot of research devoted to the use of modal logic to formalize this uncertainty faced by a group of agents in a social situation. These formal models are intended to capture both uncertainty about ground facts and uncertainty about *other* agents' uncertainty. Formal models of knowledge and beliefs have been employed by a wide range of communities, including computer scientists ([34, 110]), economists ([11, 12, 5]) and philosophers ([47]). Arguably the most successful of these frameworks are Kripke structures. Kripke structures provide a simple and well-behaved semantics for multi-agent modal logic. Despite their simplicity, there has been much discussion about whether Kripke structures are appropriate formal models of *social* situations. Much of the discussion centers around the so-called logical omniscience problem. See [71] and [34] chapter 9 for more information. From the social software point of view, the major drawback to using Kripke structures is the fact that they represent a static view of a situation. In fact, as soon as one tries representing the dynamic nature of many social situations, one of the major benefits of using Kripke structures - their simplicity - is lost.

This chapter presents a mathematical model in which the uncertainty of agents about a social situation can be represented. The next section presents the formal details of the basic model. Section 2.2 shows how this basic framework can be extended to provide a model for multi-agent epistemic temporal logics. Finally, Section 2.3 shows how to

view this framework as an extensive game, thus capturing agents' preferences in a social situation.

2.1 History Based Structures

The history based structures describe in this section have been used by a number of different communities (perhaps with additional assumptions) to reason about multi-agent interactive situations. In particular, there is an extensive discussion in the distributed computing literature of history based structures, often called *interpreted systems* (see [34] Chapters 4, 5 and 8 for a thorough discussion). The framework described in this chapter is based on that of Parikh and Ramanajam [82, 83].

Let E be a fixed set of events. As discussed in the previous section, it is natural to assume that different agents are aware of different events. To that end, assume for each agent $i \in \mathcal{A}$, a set $E_i \subseteq E$ of events "seen" by agent i. We need some notation: Given any set X (of events), X^* is the set of finite strings over X and X^ω the set of infinite strings over X. A **global history** is any sequence, or string, of events, i.e., an element of $E^* \cup E^\omega$. Let $h, h', \ldots$ range over E^* and $H, H', \ldots$ range over $E^* \cup E^\omega$. A **local history** for agent i is any element $h \in E_i^*$. Notice that local histories are always assumed to be finite.

Given two histories H' and H, write $H \preceq H'$ to mean H is a *finite* prefix of H'. Let hH denote the concatenation of finite history h with possibly infinite history H. If H is

infinite or length greater than or equal to $t \in \mathbb{N}$, let H_t denote the finite prefix of H of length t. For a history H, let $\mathsf{len}(H)$ denote the length of H (i.e., the number of events in H). For any set of histories $\mathcal{H}$, we denote the set of all histories (from $\mathcal{H}$) of length k by $\mathcal{H}_k$. Finally, define $\mathsf{FinPre}(\mathcal{H}) = \{h \mid h \in E^*, h \preceq H, \text{ and } H \in \mathcal{H}\}$. So $\mathsf{FinPre}(\mathcal{H})$ is the set of finite prefixes of elements of $\mathcal{H}$.

A set $\mathcal{H} \subseteq E^* \cup E^\omega$ is called a **protocol** provided $\mathcal{H}$ is closed under the FinPre function, i.e., $\mathsf{FinPre}(\mathcal{H}) \subseteq \mathcal{H}$. Intuitively, the protocol is the set of possible histories that could arise in a particular social situation. Notice that for a protocol $\mathcal{H}$, the set of finite histories in $\mathcal{H}$ is equal to $\mathsf{FinPre}(\mathcal{H})$. Following [82, 83], no structure is placed on the set $\mathcal{H}$. I.e., the protocol can be *any* set of histories closed under finite prefixes. Notice that this differs from standard usage of the term protocol which is taken to mean a procedure executed by a group of agents. Certainly any procedure will generate a set of histories, but not every set of histories will be generated by some procedure. For example, suppose we consider a protocols that satisfy a *fairness* property. That is every history that contains a request event (say e_r) always contains an answer (say e_a) in some finite amount of time. It is not hard to see that if we take a protocol generated by a procedure to be the set of *all* possible generated histories, then $\mathcal{H} = \{H \mid H = H_1 e_r H_2 e_a H_3$ where H_1 and H_2 are finite histories$\}$ cannot be generated by any procedure. For if a procedure can generate all histories of the form $H_1 e_r H_2 e_a H_3$ where the length of H_2 can be *any* finite number, then the procedure can also procedure

a string of the form $H_1 e_r H_2$ where H_2 does not contain e_a.

Definition 1 *Given a set of events E and a finite set of agents $\mathcal{A}$, a* **history based multi-agent structure** *based on E is a tuple $\langle \mathcal{H}, E_1, \ldots, E_n \rangle$, where $\mathcal{H} \subseteq E^* \cup E^\omega$ is a protocol and $E_i \subseteq E$ for each $i \in \mathcal{A}$.*

Single agent history based structures have been successfully used by computer scientists to reason about computational procedures and reactive systems. The main idea is that given a computational procedure, which can be represented by a finite state transition system, a history represents a possible sequence of states that can be generated as the program executes. This has led to the development of modal logics which can be used to reason about these structures. For example, the language of LTL, *linear temporal logic*, includes formulas of the form $\bigcirc\phi$, which is intended to mean that ϕ holds at *the* next moment. Notice that this assumes that there is a unique next event to take place, hence the name *linear* temporal logic. The next section contains the formal details about LTL. Other languages such as CTL and CTL^* can be used to reason about nondeterministic computational procedures, where there may not be a unique next event. The reader is referred to [107, 48] for information on temporal logic and [24] for its uses in computer science.

From the social software point of view, multi-agent history based structures provide means by which we can describe and study many important aspects of social interactions. The main idea is that each $i \in \mathcal{A}$ is only "aware" of the events $e \in E_i$. A global history

H represents a sequence of events that have taken place and each agent i may or may not be aware of the entire sequence H. This will be made more formal below. There are two things that are important to realize about multi-agent history based structures at this point. The first is that if an agent is aware of event e, this does not necessarily mean that the agent performed an action which *caused* the event to take place. In general, there may be a subset $A_i \subseteq E_i$ of events, called *actions*, that an agent can cause. This is discussed in Section 2.3 below. The second thing which is important to realize at this point is that a history based multi-agent structure is a very low-level description of a social situation. It is similar to describing a computation using machine code. Thus, it should not be surprising that many features of social situations relevant for the analysis of social procedures can be captured by these models. Whether these aspects of social situations can be captured with elegant formalisms amenable to human and/or computer analysis is another issue all together.

2.2 Logics of Knowledge and Time

In this section, we show how history based structures defined in the previous section can be used to generate models for temporal epistemic logics. As discussed above, given a particular finite global history H and an agent i, i will only "see" the events in H that are from E_i. In other words, from agent i's point of view at any time t, the initial segment H_t of H looks as if it is some sequence in E_i^*. Formally, we define a local view

function λ_i for each agent i, where $\lambda_i(H) \in E_i^*$ is agent i's *view* of history H.

Definition 2 *Let $\mathcal{H}$ be a protocol. For each $i \in \mathcal{A}$ call any function $\lambda_i : \mathsf{FinPre}(H) \rightarrow E_i^*$ a* **local view function** *of agent i.*

Note that in the above definition, λ_i is *any* function from finite strings of events to the set of i's local histories. However, we may want to place some conditions on the local view functions that we consider. The first condition assumes that an agent's local clock is "consistent" with the global clock.

- For all $H \in \mathcal{H}$ if $t \leq m$, then $\lambda_i(H_t) \preceq \lambda_i(H_m)$

This is a very natural assumption and for this thesis we will assume that all local view functions satisfy this condition. A second condition that we may want to place on the local view functions is that $\lambda_i(H)$ is *embeddable*[1] in H. Informally, this means that the agents are not wrong about the events that they witness. Finally, note that the domain of the local view functions are the *finite* strings of $\mathcal{H}$. This is in line with the assumption that at any moment only a finite number of events have already taken place. This assumption can be dropped and the definitions can be modified to allow agents the ability to remember an infinite number of events, but since our intended application is the analysis of social procedures and procedures typically have a starting point, we will stay with this more realistic assumption.

[1] A string w is embeddable in v if each character from w appears in v in the same order. For a formal definition of embeddable refer to [79]. For example, the string *abc* is embeddable in *aabbaaca*, but *abc* is not embeddable in *bbaac*.

Let H and H' be two global histories in some protocol $\mathcal{H}$. We write $H \sim_i H'$ if according to agent i, H is 'equivalent' to H'. Formally, this equivalence relation is defined in terms of the local view functions:

Definition 3 *Let $\mathcal{H}$ be a protocol. Given finite global histories, $H, H' \in \mathcal{H}$, we say that H and H' are equivalent for agent i, written $H \sim_i H'$, iff $\lambda_i(H) = \lambda_i(H')$.*

It is easy to see that for each $i \in \mathcal{A}$, $\sim_i$ is an equivalence relation. Thus by using local view functions to represent agent uncertainty, we are assuming an $\mathbf{S5}_n$ logic of knowledge[2]. Alternatively, if weaker multi-modal logics such as $\mathbf{S4}_n$ or $\mathbf{KD45}_n$ are[3] used to formalize the agents' knowledge or beliefs, then instead of starting with local view functions and deriving the relation $\sim_i$, one can assume a relation $\sim_i$ on $\mathcal{H}$ with the appropriate properties. Adding local view functions to a history based multi-agent structures gives us a history based multi-agent frame.

Definition 4 *Given a history based multi-agent structure for a set of agents $\mathcal{A}$, $\mathcal{F}_H = \langle \mathcal{H}, E_1, \ldots, E_n \rangle$ based on E, a* **history based frame** *based on $\mathcal{F}_H$ is a tuple $\mathcal{F}_K = \langle \mathcal{H}, E_1, \ldots, E_n, \lambda_1, \ldots, \lambda_n \rangle$, where each λ_i is a local view function.*

[2]The definition of the logical system $\mathbf{S5}_n$ will be given below.

[3]The modal logic $\mathbf{S4}_n$ contains Modus Ponens, an axiomatization of propositional calculus, the rule of necessitation (from ϕ infer $K_i\phi$), and the following axiom schemes: $K_i(\phi \to \psi) \to (K_i\phi \to K_i\psi)$, $K_i\phi \to \phi$ and $K_i\phi \to K_iK_i\phi$.

The modal logic $\mathbf{KD45}_n$ contains Modus Ponens, an axiomatization of propositional calculus, the rule of necessitation (from ϕ infer $K_i\phi$), and the following axiom schemes: $K_i(\phi \to \psi) \to (K_i\phi \to K_i\psi)$, $K_i\bot \to \bot$, $K_i\phi \to K_iK_i\phi$ and $\neg K_i\phi \to K_i\neg K_i\phi$. More information can be found in [21].

Additional assumptions about the agents' local view functions allows us to model agents with different capabilities. Two assumptions which have been discussed in the literature are **perfect recall** and its dual **no learning**. Intuitively, an agent is said to have perfect recall if it remembers every event that it sees. Informally, this implies that as time increases, the set of histories that an agent considers possible stays the same or decreases. We will only consider the assumption of perfect recall[4]. In [51], perfect recall is defined as follows:

Definition 5 *Let $\mathcal{F}_K$ be a multi-agent history based knowledge frame. Agent i is said to have* **perfect recall** *provided for each finite $H, H', H'' \in \mathcal{H}$, if $\lambda_i(H) = \lambda_i(H')$ and $H'' \preceq H$, then there is a global history $H''' \in \mathcal{H}$ such that $H''' \preceq H'$ and $\lambda_i(H'') = \lambda_i(H''')$.*

Suppose $\mathcal{H}$ is a protocol. Let $H \in \mathcal{H}$ and define $H_{i,t} = \{H' \mid \exists m \in \mathbb{N},\ H_t \sim_i H'_m\}$. Then it is easy to see that agent i has perfect recall iff for all $t \in \mathbb{N}$ and $H \in \mathcal{H}$, $H_{i,t+1} \subseteq H_{i,t}$. For example, consider the following recursive definition of a local view function for agent i. Let $\lambda_i(e) = e$ if $e \in E_i$ and the empty string otherwise. For each $He \in \mathcal{H}$,

$$\lambda_i(He) = \begin{cases} \lambda_i(H)e & e \in E_i \\ \lambda_i(H) & \text{otherwise} \end{cases}$$

[4]The intuitive interpretation of **no learning** is that as time increases the set of histories that an agent considers possible stays the same or increases. The interested reader is referred to [51] for more information.

Then it is easy to see that this function satisfies the property in Definition 5. Notice that when an agent has perfect recall, then the agent's local view function is derivable from the set E_i. That is, we can assume that λ_i is defined as above. Let $\mathcal{F}_K^{\mathbf{pr}}$ denote a history based knowledge frame in which each agent has perfect recall and $\mathbb{F}_K^{pr}$ the class of history based knowledge frames with perfect recall.

Finally, a few comments about whether agents have access to the global clock. We say that a history based knowledge frame $\mathcal{F}_K$ is **synchronous** if all agents have access to the global clock. Formally, $\mathcal{F}_K$ is synchronous iff for each $i \in \mathcal{A}$ and for each $H, H' \in \mathcal{H}$, if $\lambda_i(H_t) = \lambda_i(H'_u)$, then $t = u$. This property can be achieved by assuming there exists a special event $c \in E$ with $c \in E_i$ for each $i \in \mathcal{A}$. This event represents a clock tick. In synchronous history based models with perfect recall, the local view function maps each event seen by agent i in some finite history H to itself, and all other events to the clock tick c. We write $\mathcal{F}_K^{\mathbf{sync}}$ if $\mathcal{F}_K$ is synchronous and $\mathbb{F}_K^{sync}$ for the class of synchronous history based knowledge frames. We say that $\mathcal{F}_K$ is **asynchronous** if $\mathcal{F}_K$ is not synchronous. Another assumption which has been considered is a **unique initial state**. A history based knowledge frame has a unique initial state if each global history begins with the same event. Formally, a protocol has a unique initial state if there is an event $e \in E$ such that for each $H \in \mathcal{H}$, the first event in H is e.

Properties of history based knowledge frames can be described by a multi-modal logic. Let At be a countable set of propositional variables. A formula of multi-agent

knowledge and time (denoted $\mathcal{L}_n^{KT}(\mathsf{At})$ or $\mathcal{L}_n^{KT}$ if At is given) has the following syntactic form

$$\phi := p \mid \neg\phi \mid \phi \wedge \phi \mid K_i\phi \mid \bigcirc\phi \mid \phi\, U\phi$$

where $p \in \mathsf{At}$ and $i \in \mathcal{A}$. Let $\vee, \rightarrow, L_i$ be defined as usual, $\bot$ denote $p \wedge \neg p$ and $\top$ denote $\neg\top$. $K_i\phi$ is intended to mean that i "knows ϕ". $\bigcirc\phi$ is intended to mean "ϕ is true after the next event" and $\phi\, U\psi$ is intended to mean that "ϕ is true until ψ becomes true". Other temporal operators can be defined as usual. For example, $F\phi$ which is intended to mean that "ϕ will be true sometime in the future" is definable using the U operator: define $F\phi$ to be $\top\ U\ \phi$. Then "ϕ is true at every moment in the future", denoted $G\phi$, is defined to be $\neg F \neg\phi$.

Definition 6 *Suppose that $\mathcal{A}$ is a set of agents and $\mathcal{F}_K = \langle \mathcal{H}, E_1, \ldots, E_n, \lambda_1, \ldots, \lambda_n \rangle$. A* **history based model of knowledge and time** *based on $\mathcal{F}_K$ is a tuple $\mathcal{M}_H = \langle \mathcal{H}, E_1, \ldots, E_n, \lambda_1, \ldots, \lambda_n, V \rangle$, where $V : \mathsf{FinPre}(\mathcal{H}) \rightarrow 2^{\mathsf{At}}$ is a valuation function.*

For simplicity we begin by defining truth in models based on synchronous history based frames. Formulas are interpreted at pair H, t where $H \in \mathcal{H}$ is an infinite **global** history and $t \in \mathbb{N}$. That is, for $H \in \mathcal{H}$, $H, t \models \phi$ is intended to mean that in history H at time t, ϕ is true. Truth is defined recursively on the structure of a formula ϕ. Let $\mathcal{M}_H = \langle \mathcal{H}, E_1, \ldots, E_n, \lambda_1, \ldots, \lambda_n, V \rangle$ be a history based model, H an infinite global

history and $t \in \mathbb{N}$.

1. $H, t \models p$ iff $p \in V(H_t)$

2. $H, t \models \phi \wedge \psi$ iff $H, t \models \phi$ and $H, t \models \psi$

3. $H, t \models \bigcirc\phi$ iff $H, t+1 \models \phi$

4. $H, t \models \phi\ U\psi$ iff there exists $m \geq t$ such that $H, m \models \psi$ and for all l such that $t \leq l < m$, $H, l \models \phi$

5. $H, t \models K_i\phi$ iff for all $H' \in \mathcal{H}$ such that $H_t \sim_i H'_t$, $H', t \models \phi$.

We have two remarks about the above definitions. The first is that in the above definition of truth of the K_i modality (item 5. above), it is assumed that the agents all share a global clock. Thus only histories of the same length need to be considered. This assumption is made in order to simplify the presentation. If the agents do not share a global clock, then item 5. should be replaced with the following definition:

5'. $H, t \models K_i\phi$ iff for all $m \geq 0$, for all $H' \in \mathcal{H}$ such that $H_t \sim_i H'_m$, $H', m \models \phi$

The second remark concerns the definition of the U operator. It is well known that if we replace 4. with the more general 4'. below, then we can define $\bigcirc\phi$ as $\bot U\phi$.

4'. $H, t \models \phi\ U\psi$ iff there exists $m > t$ such that $H, m \models \psi$ and for all l such that $t < l < m$, $H, l \models \phi$

However, we have opted to stick with the less general definition of U (4.) to ease exposition.

Given a history based knowledge model $\mathcal{M}_H$, we say ϕ is valid in $\mathcal{M}_H$, denoted $\mathcal{M}_H \models \phi$, if for each $H \in \mathcal{H}$ and $t \in \mathbb{N}$, $H, t \models \phi$. We say ϕ is valid in a history based knowledge frame $\mathcal{F}_K$, written $\mathcal{F}_K \models \phi$, if ϕ is valid in every model based on $\mathcal{F}_K$.

Notice that we are only interpreting formulas at *infinite* global histories. This is because the definition of truth of $\bigcirc\phi$ may not make sense if the global history is finite. That is if $\mathsf{len}(H) = k$, then how should we interpret $H, k \models \bigcirc\phi$? It is easy to see that specifying that $\bigcirc\phi$ is always true (or always false) conflicts with axiom $T2$ below.

A sound and complete axiomatization for knowledge and time under various assumptions can be found in [51], using a slightly different framework. The precise connection between the two frameworks will be discussed below. We first report the relevant results from [51]. For reasoning about knowledge alone at a fixed moment in time, the following axiom system is well known to be sound and complete with respect to the class of all history based frames (see [34] for a proof).

PC. All tautologies of propositional logic

K2. $K_i(\phi \rightarrow \psi) \rightarrow (K_i\phi \rightarrow K_i\psi)$

K3. $K_i\phi \rightarrow \phi$

K4. $K_i\phi \rightarrow K_iK_i\phi$

K5. $\neg K_i\phi \rightarrow K_i\neg K_i\phi$

MP. From ϕ and $\phi \rightarrow \psi$ infer ψ

N. From ϕ infer $K_i\phi$

Call this axiom system $\mathbf{S5}_n$. The following axiom system is from [51] is used to reason about (linear) time.

T1. $\bigcirc\phi \wedge \bigcirc(\phi \rightarrow \psi) \rightarrow \bigcirc\psi$

T2. $\bigcirc(\neg\phi) \leftrightarrow \neg \bigcirc \phi$

T3. $\phi\ U\psi \leftrightarrow \psi \vee (\phi \wedge \bigcirc(\phi\ U\psi))$

RT1. From ϕ infer $\bigcirc\phi$

RT2. From $\phi' \rightarrow \neg\psi \wedge \bigcirc\phi'$ infer $\phi' \rightarrow \neg(\phi\ U\psi)$

A few remarks about the rule *RT*2. This rule is equivalent to the following simpler two rules:

RT2_1. From $\phi_1 \rightarrow \psi_1$ and $\phi_2 \rightarrow \psi_2$ infer $(\phi_1 U \psi_2) \rightarrow (\phi_2 U \psi_2)$

RT2_2. From $\bigcirc\phi \rightarrow \phi$ infer $F\phi \rightarrow \phi$

To see that RT2 follows from these rules, suppose that we have derived $\phi' \rightarrow \neg\psi$ and $\phi' \rightarrow \bigcirc\phi'$. Then, using standard propositional reasoning and T2 we can infer $\bigcirc\neg\phi' \rightarrow$

$\neg\phi'$. Hence using $\mathrm{RT2}_2$ we can infer $F\neg\phi' \rightarrow \neg\phi'$, i.e., $(\top U\neg\phi') \rightarrow \neg\phi'$. Now notice that for any formula ϕ, we can derive $\phi \rightarrow \top$ and using propositional reasoning we can infer $\psi \rightarrow \neg\phi'$. Thus using $\mathrm{RT2}_1$, we can infer $(\phi U\psi) \rightarrow (\top U\neg\phi')$; and so using propositional reasoning we can conclude $\neg(\phi U\psi)$. Showing $\mathrm{RT2}_1$ and $\mathrm{RT2}_2$ follow form RT2 is straightforward exercise in Hilbert style derivations and so will be omitted.

Call the axiom system that contains the rules and axiom schemes from $\mathbf{S5}_n$ and the rules and axiom schemes above $\mathbf{S5}_n^U$. Again it is well-known that $\mathbf{S5}_n^U$ is sound and complete with respect to the class of all history based knowledge frames. It becomes much more interesting when axiom schemes connecting the knowledge and time modalities are added. The two axiom schemes from [51] that will be of interest are:

KT1. $K_i \bigcirc \phi \rightarrow \bigcirc K_i\phi$

KT2. $K_i\phi_1 \wedge \bigcirc(K_i\phi_2 \wedge \neg K_i\phi_3) \rightarrow L_i((K_i\phi_1)\ U[(K_i\phi_2)\ U\neg\phi_3])$

These axiom schemes characterize systems in which all agents are assumed to have perfect recall. Axiom $KT1$ is easily seen to be valid in synchronous history based knowledge frames with perfect recall. For if agent i knows (at the current moment) that ϕ will be true at the next moment, then since i has perfect recall, i cannot lose this knowledge. Therefore, at the next moment agent i will know ϕ. Using similar reasoning, the formula $K_iG\phi \rightarrow GK_i\phi$ – if i knows ϕ is true at the current moment and that it will always be true, then it will always be the case that agent i knows ϕ – is easily seen to be valid. Interestingly, van der Meyden showed that adding only this axiom to $\mathbf{S5}_n^U$ is not

complete for frames with perfect recall [103]. In [51], a series of completeness proofs are offered under a variety of assumptions (perfect recall, no learning, synchronous, unique initial state). In particular, they show that the more complicated axiom $KT2$ is what is needed to characterize frames in which the agents are assumed to have perfect recall, i.e., the axiom system $\mathbf{S5}_n^U + KT2$ is sound and complete with respect to frames with perfect recall. For a proof of these results (with respect to the semantics in [51]) refer to [51].

At this point the reader may have noticed that we have omitted discussion of a topic widely discussed in the epistemic logic literature — common knowledge. In part, this is due to an early result of Halpern and Vardi [52] who show that the complexity of reasoning about the validity problem of languages with common knowledge in frames with perfect recall is Π_1^1 complete. Hence, no recursive axiomatization is possible when common knowledge is added to our language. The result is true regardless of whether agents have access to the global clock. Only if we drop all assumptions on the reasoning abilities of the agents do we get the possibility of a finite axiomatization (or make the drastic assumptions of no learning, perfect recall, synchronous and a unique initial state). In fact, when more than one agent is involved, then reasoning about the validity problem of the axiom systems discussed in this chapter is in nonelementary time (consult [52] for proofs of this and related results).

Another point worth mentioning is that the language $\mathcal{L}_n^{KT}$ is not expressive enough to capture the synchronous property (nor the unique initial state property). The completeness proof for $\mathbf{S5}_n^U$ holds regardless of whether the history based knowledge frames are assumed to be synchronous. These properties can be captured in languages with past-time operators. Completeness results for such systems have recently been established in [40]. Other interesting properties of history based frames cannot be captured in our language, such as that ϕ is true at the next moment in *all* possible extensions of the current history. In [104], van der Meyden and Wong prove a series of completeness results for logics of knowledge with branching time operators.

2.2.1 Histories or Runs?

The section discusses the similarities and differences between the Parikh and Ramanajam framework described in this chapter and the Halpern et al. [52, 34, 51] interpreted systems.

We begin by formally defining interpreted systems. The reader is referred to [34] and [51] for more details. Let L be a set of local states. A **system** for n agents is a set $\mathcal{R}$ of **runs**, where a run $r \in \mathcal{R}$ is a function $r : \mathbb{N} \rightarrow L^{n+1}$ $r(t)$ has the form $\langle l_e, l_1, \ldots, l_n \rangle$, where l_e is the state of the environment, l_i for $i = 1, \ldots, n$ is the local state of each agent. A **point**, or global state, is an element $(r, t) \in \mathcal{R} \times \mathbb{N}$. An **interpreted system** $\mathcal{I} = (\mathcal{R}, \pi)$, where $\mathcal{R}$ is a system and $\pi : (\mathcal{R} \times \mathbb{N}) \times \mathsf{At} \rightarrow \{\mathbf{true}, \mathbf{false}\}$, that is $\pi(r, t)$ is

a truth assignment, where At is the set of atomic propositions. The uncertainty of the agents is defined as follows: agent i cannot distinguish two points if it is in the same state in both: $(r,t) \sim_i (r',t')$ iff $r(t)_i = r'(t')_i$. Formulas are interpreted at pairs (r,t) where $r \in \mathcal{R}$ and $t \in \mathbb{N}$, i.e., $r,t \models \phi$ is intended to mean that in run r at time t ϕ is true. The formal definition of truth is very similar the the definition above, and so we will only give the definition of the modal operators (see [51] for more details). Let $\mathcal{I} = (\mathcal{R}, \pi)$ be an interpreted system, $r \in \mathcal{R}$ and $t \in \mathbb{N}$. Then

1. $r,t \models K_i\phi$ iff $r',t' \models \phi$ for all (r',t') such that $(r,t) \sim_i (r',t')$
2. $r,t \models \bigcirc\phi$ iff $r,t+1 \models \phi$
3. $r,t \models \phi U \psi$ iff there is some $t' \geq t$ such that $r,t' \models \psi$, and for all l with $t \leq l < t'$, we have $r,l \models \phi$

At first glance, the difference between an interpreted system and a history based model seems to be purely linguistic. A run is a function that specifies the local state of each agent (including the environment), but can just as easily be understood as a sequence of events, where each event is a tuple of local states. For Parikh and Ramanajam, an event is a primitive object, whereas for Halpern et al. events are tuples of local states. We make this observation more formal below.

We first discuss the translation from history based models to interpreted systems. Let $\mathcal{F}_K = \langle \mathcal{H}, E_1, \ldots, E_n, \lambda_1, \ldots, \lambda_n \rangle$ be a history based knowledge frame based on E and

$\mathcal{M}_H = \langle \mathcal{H}, E_1, \ldots, E_n, \lambda_1, \ldots, \lambda_n, V \rangle$ a model based on $\mathcal{F}_K$. We define an interpreted system $\iota(\mathcal{M}_H) = (\mathcal{R}_\mathcal{H}, \pi)$ as follows.

Let $L = \bigcup_{i \in \mathcal{A}} \{\lambda_i(H_t) \mid H \in \mathcal{H}, t \in \mathbb{N}\}$. Let e denote the environment agent and assume that for each finite history $H \in \mathcal{H}$, $\lambda_e(H) = H$. That is, the environment agent is aware of every event (this is only for convenience). For each infinite $H \in \mathcal{H}$ define a run $r_H : \mathbb{N} \to L^{n+1}$ as follows: $r_H(t) = \langle \lambda_e(H_t), \lambda_1(H_t), \ldots, \lambda_n(H_t) \rangle$. Then the following observation is a straightforward application of the definition.

Observation 2.2.1 *For each infinite history $H, H' \in \mathcal{H}$, for each $t, m \in \mathbb{N}$, $H_t \sim_i H'_m$ iff $(r_H, t) \sim_i (r_{H'}, m)$.*

Finally, interpret the valuation function in the obvious way. That is, for each $p \in \mathsf{At}$, define $\pi(r_H, t)(p) = \mathbf{true}$ provided $p \in V(H_t)$.

Lemma 2.2.1 *Let $\mathcal{M}_H = \langle \mathcal{H}, E_1, \ldots, E_n, \lambda_1, \ldots, \lambda_n, V \rangle$ be a history based model and $\phi \in \mathcal{L}_n^{KT}$ an arbitrary formula. Then for each $H \in \mathcal{H}$ and $t \in \mathbb{N}$*

$$H, t \models \phi \text{ iff } r_H, t \models \phi$$

Proof Let $\mathcal{M}_H = \langle \mathcal{H}, E_1, \ldots, E_n, \lambda_1, \ldots, \lambda_n, V \rangle$ be a history based model of knowledge and time, $\phi \in \mathcal{L}_n^{KT}$, $H \in \mathcal{H}$ and $t \in \mathbb{N}$. We will show $H, t \models \phi$ iff $r_H, t \models \phi$. The proof is by induction on ϕ. The base case is true by definition. The boolean cases are obvious which leaves the modal cases.

- Suppose that $H, t \models K_i\phi$, then for each $m \geq 0$ and $H' \in \mathcal{H}$ with $H_t \sim_i H'_m$, $H', m \models \phi$. We must show that $r_H, t \models K_i\phi$. Let $r_{H'} \in \mathcal{R}_{\mathcal{H}}$ be any arbitrary run such that $(r_H, t) \sim_i (r_{H'}, m)$. By the above observation, $H_t \sim_i H'_m$. Since $H, t \models K_i\phi$, $H', m \models \phi$. Hence by the induction hypothesis, $r_{H'}, m \models \phi$. Therefore, $r_H, t \models K_i\phi$. The other direction is analogous.

- $H, t \models \bigcirc\phi$ iff $H, t+1 \models \phi$ iff (induction hypothesis) $r_H, t+1 \models \phi$ iff $r_H, t \models \bigcirc\phi$.

- $H, t \models \phi\ U\psi$ iff $\exists t' > t$ such that $H, t' \models \psi$ and for each m with $t < m < t'$, $H, m \models \phi$ iff (induction hypothesis) $\exists t' > t$ such that $r_H, t' \models \psi$ and for each m with $t < m < t'$, $r_H, m \models \phi$ iff $r_H, t \models \phi\ U\psi$. □

This lemma shows that the soundness results from [51] can be applied to history based frames. For example, suppose that ϕ is a theorem of $\mathbf{S5}_n^U + KT1$ but $\mathcal{F}_K \not\models \phi$. Then there is a model $\mathcal{M}_H$ based on $\mathcal{F}_K$ in which there is a global history H and moment $t \in \mathbb{N}$ such that $H, t \not\models \phi$. But then using the above lemma in the interpreted system $\iota(\mathcal{M}_H)$, $r_H, t \not\models \phi$ which contradicts the soundness proof for interpreted systems.

What about the completeness results? I.e., do the completeness results from [51] apply to history based frames? The answer is yes if we can show that for each interpreted system, there is a modally equivalent[5] history based frame.

[5]That is, the two structures satisfy the same modal formulas.

Let $(\mathcal{R}, \pi)$ be an interpreted system with local states L. Given a run $r \in \mathcal{R}$ we show how to construct a history H^r. First let $E_i = L$ for each agent i (or $E_i = L_i$ if the agents do not share local states). Then let $E = \cup_i E_i \cup L^{n+1}$. So the events are the local states and the global states. For each $r \in \mathcal{R}$, let $H^r = r(0)r(1)r(2)\cdots$. Let $\mathcal{H}' = (\cup_i E_i)^* \cup \{H^r \mid r \in \mathcal{R}\}$ and $\mathcal{H}_\mathcal{R} = \mathcal{H}' \cup \mathsf{FinPre}(\mathcal{H}')$. Notice that in $\mathcal{H}_\mathcal{R}$ the only infinite histories are histories of the form H^r for some $r \in \mathcal{R}$. Thus these are the *only* histories in which we interpret our formulas. We need only define the agents local view function. Since the domain of the local view function is the set of all finite prefixes of a protocol, we have two cases the consider. The first is if $H \in E_i^*$ for some $i \in \mathcal{A}$. Then simply define $\lambda_i(H) = H$ (as this situation will not arise when interpreting formulas, the actual value of $\lambda_i(H)$ does not matter). If H is H^r for some $r \in \mathcal{H}$, then define $\lambda_i(H^r) = \mathsf{last}(H^r)_i$. Then the following observation is obvious.

Observation 2.2.2 *Let $\mathcal{R}$ be any set of runs, then for each $r, r' \in \mathcal{R}$ and $t, m \in \mathbb{N}$, $r(t)_i = r'(m)$ iff $H^r_t \sim_i H^{r'}_m$.*

Finally, we define a valuation function $V : \mathsf{FinPre}(\mathcal{H}) \to 2^{\mathsf{At}}$ as follows. If $H \in E_i$ for some $i \in \mathcal{A}$, then $V(H) = \emptyset$. If H is H^r for some $r \in \mathcal{R}$, then let $p \in V(H^r_t)$ iff $\pi(r, t)(p) = \mathbf{true}$. The following lemma which shows that our translation works as intended is obvious so the proof is omitted.

Lemma 2.2.2 *Let $(\mathcal{R}, \pi)$ be an interpreted system. Then for each $r \in \mathcal{R}$ and each*

$\phi \in \mathcal{L}_n^{KT}$

$$r, t \models \phi \text{ iff } H^r, t \models \phi$$

The only case that may cause some trouble is when ϕ is of the form $K_i\psi$. In this case, the results follows immediately once one notices that formulas are only interpreted at infinite histories, i.e., histories of the form H^r for some $r \in \mathcal{R}$. Thus if $H^r, t \models K_i\psi$. Then if $H^r \sim_i H'$, H' must be of the form $H^{r'}$ for some $r' \in \mathcal{R}$ (these are the only possible infinite histories). And the proof of the lemma follows immediately.

The above lemma shows that the completeness proofs from [51] carry over to history based frames. However, the above construction seems like somewhat of a cheat. In particular, notice that the local view functions are *not* embeddable in a infinite global history H^r. A better solution would be for a given interpreted system $(\mathcal{R}, \pi)$ to find a set of events E and protocol $\mathcal{H}$ based on E and *embeddable* local view functions such that the history based model based on this frame is modally equivalent to $(\mathcal{R}, \pi)$. However, answering this question is is analogous to constructing a program written in a high level language (such as C) from some machine code. In fact, the real question we are asking is *where does a particular interpreted system come from*? For more on this topic the reader is referred to [34] Chapter 5.

This section shows that the answer to the question posed in the title of this section, is that is does not really matter which semantics one chooses from the point of view of soundness and completeness of axiom systems. So, is the difference between the two

semantics only linguistic? Technically, perhaps the answer is yes. However, there is a difference from the modeler's point of view. The intuition guiding interpreted systems is that there is a computational procedure that each agent is following and the local states describe the internal states of the agents at different moments in time. So the difference lies in the intended application in the models. For interpreted systems, the intended application is an analysis of distributed computational procedures whereas for history based structures the intended application is social interactive situations. For example, in [83], Parikh and Ramanajam argue that this framework very naturally formalizes many social situations by providing a semantics of messages in which notions such as Gricean implicature can be represented.

2.3 Extensive Games

In the game theory literature, *extensive game forms* are used to model the decision problems encountered by agents in strategic situations [65]. These structures have much in common with history based structures. In fact, as will be shown in this chapter, with some additional assumption, a history based structure can be turned into an extensive game form.

Let $\langle \mathcal{H}, E_1, \ldots, E_n \rangle$ be a history based structure based on a set of events E. In a general game-theoretic situation the events are "caused" by the agents. As such, we assume for each agent a set $A_i \subseteq E_i$ of *actions i can perform.* Notice that we are blurring

the distinction between the action that agent i chooses to cause event e and the event e itself.

In this section we will assume that the agents are aware of all possible events, i.e., for each $i \in \mathcal{A}$, $E_i = E$. This assumption is called *perfect information* is the game theory literature. Actually, the assumption of perfect information says something slightly stronger, namely that each agent *knows* which actions have taken place. Essentially, this means that we are we are working with synchronous history based frames with local view functions that satisfy perfect recall. Thus to make this assumption of perfect information formal, we must bring in the machinery from Section 2.2. Reasoning about extensive games from this is point of view is explored in a recent paper of Bonanno [13]. In order not to overburden the reader with notation at this point, we will not make any attempt to bring in the local view functions from Section 2.2 and assume that we are working under the assumption of perfect information. Therefore, in this section, we will denote a history based structure as $\langle \mathcal{H}, A_1, \ldots, A_n \rangle$ and assume $E_i = E$ for all i where E is the set of all possible events.

A few structural assumptions about the protocol $\mathcal{H}$ are needed. The first assumption amounts to saying that at any moment only one agent can perform an action. Given a global history H (possibly infinite) and time $t \in \mathbb{N}$, let $\mathcal{F}(H_t) = \{H' \in \mathcal{H} \mid H_t \preceq H'\}$, i.e., $\mathcal{F}(H_t)$ is the set of all global histories from $\mathcal{H}$ that extend H_t. Also, suppose that for each global history H, $\mathsf{first}(H)$ denotes the first event of H.

Definition 7 *A protocol is said to satisfy the* **single agent** *property if for each* $H \in \mathcal{H}$ *and each* $t \in \mathbb{N}$, *there is a* unique $i \in \mathcal{A}$ *such that for each* $H' \in \mathcal{F}(H_t)$, $\mathsf{first}(H') \in A_i$.

Notice that this implies that at every moment there always is an agent who performs some action. If necessary, we can assume that nature is an agent which can always perform a special action c interpreted as a clock tick. One more technical assumption is needed in order to deal with infinite histories.

Definition 8 *A protocol* $\mathcal{H}$ *is said to be* **closed** *provided for each infinite history* H, *if for each* $t \in \mathbb{N}$, $H_t \preceq H$ *and* $H_t \in \mathcal{H}$, *then* $H \in \mathcal{H}$. *I.e.,* $\mathcal{H}$ *is closed upwards under the* $\preceq$ *relation.*

These additional structural assumptions about the protocol $\mathcal{H}$ is all that is needed to define an extensive game form.

Definition 9 *A* **history based game form** *is a structure* $\mathcal{F}_G = \langle \mathcal{H}, A_1, \ldots, A_n \rangle$ *where* $\mathcal{H}$ *is a protocol that is closed and satisfies the single agent property.*

We say a history based game form is *finite* if $\mathcal{H}$ is finite and is *finite-horizon* if each $H \in \mathcal{H}$ is finite. What are missing from the above definition are the preferences of the agent. In the game theory literature it is standard to assume that agents have preferences over possible outcomes of a situation. In other words, the players preferences are over *terminal* histories, i.e., histories that cannot be extended. Formally,

define a set $\mathsf{Term}(\mathcal{H})$ of all **terminal** histories in $\mathcal{H}$ as follows $\mathsf{Term}(\mathcal{H}) = \{H \mid H \in \mathcal{H}$ and if $H \prec H'$, then $H' \notin \mathcal{H}\}$.

Here we have a choice — preferences can either be represented using preference relations over the set of terminal histories or by utility functions. Any function $u_i : \mathsf{Term}(\mathcal{H}) \to \mathbb{R}$ is called i's **utility function**. Intuitively, for $H, H' \in \mathsf{Term}(\mathcal{H})$ and $u_i(H) \geq u_i(H')$ then i (weakly) prefers the situation H to H'. The reader is urged to consult [65] for more information on defining preferences and utility functions in extensive game forms. We will leave the discussion here as Chapter 3 will pick up where we have left off.

Chapter 3

Knowledge Based Obligation

This chapter focuses on the first of the two main issues of this thesis: developing a formal model in which an agent's choice of action (especially obligatory action) depends on its state of knowledge. Indeed one cannot reasonably be expected to respond to a problem if one is not aware of its existence. For instance a doctor cannot be expected to treat a patient unless she is aware of the fact that he is sick, and this creates a secondary obligation on the patient or someone else to inform the doctor of his situation. In other words, many obligations are situation dependent, and are only relevant in the presence of the relevant information. This creates the notion of *knowledge based obligation.* We consider both the case of an absolute obligation (although dependent on information) as well as the notion of an obligation which may be over-ridden by more relevant information. This chapter is based on joint work with Rohit Parikh and Eva

Cogan [81].

3.1 Introduction

Suppose we are given two functions α and β over some domain D. Then $\alpha \leq \beta$ iff $\forall x \in D, \alpha(x) \leq \beta(x)$, and moreover $\alpha < \beta$ iff $\alpha \leq \beta$ and $\beta \not\leq \alpha$. If some element d of D is chosen, and we are offered a choice between $\alpha(d)$ and $\beta(d)$ in dollars, clearly we will choose $\beta(d)$ even if d is unknown to us. This paradigm comes in useful in two contexts. The decision theoretic context, where D is the set of possible states of nature and α, β represent payoff functions. The other context is the game theoretic one where D represents the (already chosen but unknown to us) choices of the other players, and α, β are possible strategies for us. In this context, if $\alpha < \beta$, we will say that β dominates α and we will tend to prefer β.

Now this comparison between α and β will not be possible for us if all we are given are the ranges of α and β. For instance if $\alpha(x) = x^2$ and $\beta(x) = x$ over the unit interval [0,1], then it is indeed the case that $\alpha < \beta$ but the ranges of the two functions are the same. Moreover, the function $\gamma(x) = 1 - x$ has the same range as β, but while we do have $\alpha < \beta$ we do not have $\alpha < \gamma$. So just knowing the ranges we could not tell that $\alpha < \beta$.

These considerations have relevance to the situation where the values represent some societal good and we ought to do what is best for society. Clearly, knowing the *set* of

consequences of action α versus knowing the set of consequences of β will not always tell us how to decide. Rather we need to ask, *given* the current circumstances (possibly only partially known to us) can we still choose? It has been suggested that action α is preferable to action β if *all* consequences of α are better than any consequence of β. But clearly this requirement is too strict.

For consider the decision whether to exercise. Suppose some people are rich and some are poor, but all would be better off exercising. However, assume for a moment that it is better to be rich and lazy than to be poor and to exercise. Then the consequences of exercising are {rich$\wedge$exercised, poor$\wedge$exercised} whereas the consequences of being lazy are {rich$\wedge$lazy, poor$\wedge$lazy}. Not *all* consequences of exercising are better than every consequence of being lazy, even though *each* individual person, whether rich or poor, is better off exercising. To ask that *all* consequences of exercising be better than every consequence of being lazy, is too much. So we need to compare situations pairwise, a particular situation with exercising and the "same" situation with laziness. In other words, if choosing between an α and a β, we should choose β if for some *specific circumstance* β yields a higher value than α.

What if it is the case that sometimes $\alpha(x) < \beta(x)$ and sometimes the other way? In that case we may need to find out more about x before we can choose. Sometimes someone else knows enough about x for this purpose and then she ought to tell us, in case she knows that we are about to choose.

The following examples from [81] illustrate the type of situations we have in mind.

Example 1: Jill is a physician whose neighbour is ill. Jill does not know and has not been informed. Jill has no obligation (as yet) to treat the neighbour.

Example 2: Jill is a physician whose neighbour Sam is ill. The neighbour's daughter Ann comes to Jill's house and tells her. Now Jill does have an obligation to treat Sam, or perhaps call in an ambulance or a specialist.

Example 3: Mary is a patient in St. Gibson's hospital. Mary is having a heart attack. The caveat which applied in case a) does not apply here. The hospital has an obligation to *be aware* of Mary's condition at all times and to provide emergency treatment as appropriate.

Example 4: Jill has a patient with a certain condition C who is in the St. Gibson's hospital mentioned above. There are two drugs d and d' which can be used for C, but d has a better track record. Jill is about to inject the patient with d, but unknown to Jill, the patient is allergic to d and for this patient d' should be used. Nurse Rebecca is aware of the patient's allergy and also that Jill is about to administer d. It is then Rebecca's obligation to inform Jill and to suggest that drug d' be used in this case.

In all the cases we mentioned above, the issue of an obligation arises. This obligation is circumstantial in the sense that in other circumstances, the obligation might not apply. Moreover, the circumstances may not be fully known. In such a situation, there may still be enough information about the circumstances to decide on the proper course of action. If Sam is ill, Jill needs to know that he is ill, and the nature of the illness, but not where Sam went to school.

Our purpose in this chapter is to set forth a framework which can be used to study situations similar to those in the four examples above and to point out certain logical properties which will hold. We take as our starting point the history based frames described in the previous chapter. Our goal is a semantics and an axiomatic system in which we can formalize the agent's reasoning in the above examples. In particular, we should be able to *formally prove* that Ann is obliged to send a message to Jill in example 2 (given the appropriate assumptions). In fact, this has been one of the goals of standard *deontic logic*. See [60, 57] and references therein for an up to date discussion of deontic logic. One of the main points discussed above is that Jill's obligation arises only after she learns of her neighbor's illness. In other words, her obligation depends on her having the appropriate *knowledge*. In much of the deontic logic literature, an agent's knowledge is only informally represented or the discussion is focused on representing epistemic obligations, i.e., what an agent 'ought to know' (see [63] for a recent discussion). The

logic in this chapter is intended to capture the *dependency* of individual obligation on knowledge.

The above discussion and examples point to four issues that are relevant to the task at hand.

1. The formal language and semantics must have machinery in which we can express statements of the form "after agent i performs action a...".

2. The formal language and semantics must have machinery in which we can express statements of the form "agent i is obliged to perform action a" or "after performing the obligatory action a....".

3. Certain actions are obligatory *only* in the presence of relevant information.

4. Certain obligations may disappear in the presence of relevant information (for example, in example 4, Jill's obligation to administer drug d disappears in the presence of relevant information).

Each of the above issues (except perhaps the third) have been the focus of much discussion in a variety of contexts. Certainly the notion of obligation has been widely discussed among philosophers, logicians and more recently computer scientists. See [57] for a survey of the literature. What is new in this analysis is an explicit representation of the dependency of an agent's obligations on its knowledge. In essence, this chapter can be described as an attempt to combine two distinct research areas: deontic logic and

epistemic logic. A complete survey of the literature relevant to each of the four issues above would require a book length treatment and would distract us from the task at hand. Instead, we point to a few references which are relevant to our formal treatment. Of course, since the epistemic part of our semantics is just the history based frames from Chapter 2, much of the discussion of the literature from that chapter is relevant here. For a treatment of obligatory actions in social situations, the reader is referred to [60]. In [60], using so-called '*see to it that*' modal operators, Horty shows how to represent obligatory actions. The next sections will discuss each of the four issues in detail.

The next three sections discuss actions, obligations and default obligations respectively. For these sections, let $\mathcal{M}_H = \langle \mathcal{H}, E_1, \ldots, E_n, \lambda_1, \ldots, \lambda_n, V \rangle$ be a fixed history based model based on a set of events E. For simplicity we will assume perfect recall and that all agents have access to the global clock. Finally, we conclude by returning to the examples discussed in this introduction and show how our framework deals with each example.

3.2 Actions

We think of an action as something which is performed at a *finite* global history H and which yields a set $a(H)$ of global extensions of H (provided that the action a *can* be performed at H). In general there will be *other* extensions of H in which a has not been performed. Formally, we assume a finite set, Act, of actions that is a subset of E (the

set of possible events). We assume that each action is tagged to a particular agent who is the only one performing that action. Thus if l stands for *turning on the light*, then l_j will be *Jill turning on the light*, and l_s will be *Sam turning on the light.* Clearly Jill cannot perform the action l_s. For the *sake of simplicity* we assume that at any moment of time *only* one agent can perform any action, although if that agent does nothing, then nature is free to perform a clock tick. In other words, we assume that the protocol satisfies the single agent property (Definition 7).

Formally, we assume that the set of actions $\mathsf{Act} \subseteq E$ is partitioned into sets Act_i for each agent $i \in \mathcal{A}$. That is, $\mathsf{Act} = \cup_{i \in \mathcal{A}} \mathsf{Act}_i$ where $\mathsf{Act}_i \cap \mathsf{Act}_j = \emptyset$ for $i \neq j$. Elements of Act_i will be denoted by a_i, b_i, etc. If it is clear from context which agent can perform which action, then we will leave out the indices.

We understand an action $a \in \mathsf{Act}$ as a partial function from the set of finite histories to sets of global histories. If Ann turns on the light at H_t then the corresponding set is the set of all histories H' such that H' extends $H_t l_a$. Formally, given a an infinite global history H and a time $t \in \mathbb{N}$:

$$a(H_t) = \{H' \mid H_t a \preceq H' \text{ and } H' \in \mathcal{H}\}$$

This implies that when an action is performed, it is performed at the next moment of time. We could weaken this assumption and assume that performing an action means

performing that action eventually. In this case, $a(H_t)$ will be the set of global histories H' such that there is an $H_1 \in E^*$ and $H_t H_1 a \preceq H'$. Note that in this case for two different actions a and b which both can be performed at finite global history H_t, $a(H_t)$ and $b(H_t)$ need not be disjoint. We will use either definition depending on the application – it should be clear from the context which is intended.

In order to reason about actions in our formal language, we introduce a **PDL** style modal operator. If $a \in Act$, then $[a]\phi$ is intended to mean that in all histories in which a is performed (by the appropriate agent), ϕ is true. I.e., all executions of a make ϕ true. Its dual $\langle a \rangle \phi$ will mean that after some execution of a, ϕ is true. Given a global history H and time t, we define truth of $[a]\phi$ as follows

$$H, t \models [a]\phi \text{ iff for all } H' \in a(H_t),\ H', t+1 \models \phi$$

Whereas the $\bigcirc$, F and U modal operators are linear time operators, i.e., they range over moments on a single global history, the dynamic modalities just defined are best understood as branching time operators.

Note that we are assuming that actions are primitive, i.e., an action is just an element of the set of events E. One could develop a calculus of actions, where complicated actions are built up from primitive actions using standard **PDL** style operators. We refer the reader to van der Meyden [105] for more on this topic. However, a large class

of examples, including all the ones we consider in [81] can be handled without adding the complications of a calculus of actions; and so we leave this line of reasoning for future research.

One last assumption is that each agent knows *when* it can perform an action. Thus if $H_t \sim_i H'_t$ and i can perform a_i at H_t then it can also perform a_i at H'_t. We note that if the power has been off and an agent does not know whether the electric power is back on, then the agent still knows it can perform the action 'flip the light switch', but does not know whether it can perform the action 'turn on the light'. Since we stipulate that an agent knows when it can perform an action, our notion of action will correspond to flipping the switch but not to turning on the light (unless there is no doubt that the power *is* on.) It is not too hard to see that this assumption will force the following axiom scheme to be valid:

$$\langle a_i \rangle \top \rightarrow K_i \langle a_i \rangle \top$$

3.3 Values

We move to the second issue discussed in the introduction to this chapter: formalizing an agent's obligation. The basic idea is to assign a real number to each infinite global history (called the **value** of the history) and assume that higher valued histories are "better" than lower valued histories. Notice that we are not making any attempt to explain *why* histories are assigned the values they are — that is a job for an ethicist (or

perhaps a court). We are interested in formalizing the agents' reasoning about obligatory actions *given an assignment of values.* Furthermore, it is worth pointing out that the actual values assigned to histories do not matter — it is only the induced linear ordering among global histories which will be of interest for us. At this stage, the use of real numbers eases presentation and suggests parallels with a game theoretic analysis. The basic idea is that our models can be thought of as extensive games in which all agents are playing the *same* utility function, or at least each agent's utility function induces the same linear order over global histories.

Under natural assumptions, (e.g. that the set of values is finite or compact) there will be a set of extensions of H which have the highest possible value. We will refer to this set as the H-**good** histories and denote it as $\mathcal{G}(H)$.

Now, since all global histories have a value, so will those global histories which extend some finite history H in which a has been performed. We will say that a **is good** to be performed at a finite history H, if $\mathcal{G}(H) \subseteq a(H)$, i.e., there are no H-good histories which do not involve the performing of a. And we say that a **may** be performed at H if $\mathcal{V}(H) \cap a(H)$ is non-empty.[1]

[1]Note that this definition seems compatible with the inference that if a letter may be posted then it may be posted or burned. But we can avoid this apparent paradox by saying that the permission to post or burn a letter really amounts to a permission to post the letter plus the permission to burn it. This can be formally expressed as the formula, $(\mathcal{G}(H) \cap (a(H) \neq \emptyset) \wedge (\mathcal{G}(H) \cap (b(H) \neq \emptyset)$ rather than the more obvious interpretation $(\mathcal{G}(H) \cap (a(H) \cup b(H)) \neq \emptyset)$ which does not justify burning the letter as an option. Here, of course, a is the action of posting the letter and b is the action of burning it. The formula $(\mathcal{G}(H) \cap a(H) \neq \emptyset)$ expresses permission to post the letter. It does imply $(\mathcal{G}(H) \cap (a(H) \cup b(H)) \neq \emptyset)$ but, in our view, the latter formula does not express the intent of the English sentence "You may post the letter or burn it."

We now make the above discussion more formal, but first some notation. Let $\mathcal{H}$ be a protocol and $H \in \mathcal{H}$ an infinite global history. Recall that for each $t \in \mathbb{N}$, $\mathcal{F}(H_t) = \{H' \in \mathcal{H} \mid H_t \preceq H'\}$. That is, $\mathcal{F}(H_t)$ is the "fan" of global histories (in $\mathcal{H}$) that contain H_t as an initial segment. Let $\mathcal{K}$ be any set of histories, $f : \mathcal{K} \to \mathbb{R}$ be any function, and define $f[\mathcal{K}] = \{f(H) \mid H \in \mathcal{K}\}$.

Definition 10 *Let $\mathcal{H}$ be any protocol. A function* $\mathsf{val} : \mathsf{Term}(\mathcal{H}) \to \mathbb{R}$ *is called a* **value function** *if for each infinite global history $H \in \mathcal{H}$,*

1. *For all $t \in \mathbb{N}$, $\mathsf{val}[\mathcal{F}(H_t)]$ is a closed and bounded subset of $\mathbb{R}$.*

2. $\cap_{t\in\mathbb{N}}\mathsf{val}[\mathcal{F}(H_t)] = \{\mathsf{val}(H)\}$

Condition 2 is a 'discounting' condition which ensures that values of histories depend only on what happens in a finite amount of time. If two histories agree for a long time then their values should be close. Formally, it is easy to see that condition 2 implies the following fact:

$$\forall H' \in \mathcal{H},\ \forall \epsilon > 0,\ \exists t \geq 0,\ \forall H' \in \mathcal{H},\ (H'_t = H_t \Rightarrow |\mathsf{val}(H'_t) - \mathsf{val}(H_t)| < \epsilon)$$

Since $\mathsf{val}[\mathcal{F}(H_t)]$ is closed and bounded for all t, there are maximal and minimal elements. Thus we define, $\mathcal{G}(H_t) = \{H' | H' \in \mathrm{argmax}(\mathsf{val}[\mathcal{F}(H_t)])\}$. Thus $\mathcal{G}(H_t)$ is the set of maximally good, (or just maximal) extensions of H_t.

We can now define knowledge based obligation.

Definition 11 *Agent i is* obliged *to perform action a at global history H and time t iff a is an action which i (only) can perform, and i* knows *that it is good to perform a. Formally, $(\forall H')(H_t \sim_i H'_t$ and $H' \in \mathcal{G}(H'_t) \Rightarrow H' \in a(H'_t))$. Putting this in terms of the agent's local history $h = \lambda_i(H_t)$, all maximal extensions of* any *H'_t with $\lambda_i(H'_t) = h$ belong to the range of the action a.*

Note that in our semantics at any moment, only one action attached to a particular agent is good. In theory nothing prevents it from being food that Ann puts the tea-kettle on the stove while Jill is treating her father, but we prefer not to overburden an already heavy semantics.

3.3.1 Comparison with Horty

This above definition of a good action generalizes Horty's notion of dominance of actions ([60]). In [60] actions are sets of global histories and at any moment m an agent i is faced with a set $Choice_i^m$ of possible actions. This set is a partition of the possible global histories that extend a global history at a particular moment m. Each history H is assumed to have a value $Value(H)$. Since actions are in fact sets of global histories, one is tempted to compare actions pointwise so that action a is 'better' that a' just in case $Value(H) \geq Value(H')$ for each $H \in a$ and $H' \in a'$. In such a case we will write

$a \geq a'$ ($\leq$, $<$, $>$ can then be defined in similar ways). However, using the *sure-thing principle* of Savage, Horty demonstrates some problems with this definition. In order to get around this complication, actions are given a functional flavor.

For each agent i and moment m let $State_i^m$ be the actions available to each agent other than i. Thus $State_i^m$ is a collection of actions available to agent i which are themselves sets of global histories. That is

$$State_i^m = Choice_{\mathcal{A}-\{i\}}^m$$

where $\mathcal{A}$ is the set of all agents[2]. Horty can now compare actions as follows (recall that actions are defined to be sets of global histories)

Definition 12 (Horty [60]) *Let i be an agent, m a moment and a and a' be two members of $Choice_i^m$. Then (a'* weakly dominates *a) $a \preceq a'$ if and only if $a \cap s \leq a' \cap s$ for each $s \in State_i^m$; and $a \prec a'$ if $a \preceq a'$ and not $a' \preceq a$.*

Thus when comparing actions a and a', they are treated as functions over the domain of choices of the other agents (i.e., the domain is $State_i^m$). As functions, a and a' are then compared pointwise. Our approach is to make this idea explicit and define actions as partial functions on the set of all possible histories. We then can compare actions

[2]We have only defined the set $Choice_i^m$ for one agent, so the above definition only makes sense if there are only two agents. However, this definition can be extended to multiple agents, see [60] for more details.

pointwise on their domains.

3.4 Default Histories

As we have already seen from example 4, the notion of a *default* history is important for our analysis. Since the notion of obligation in this chapter depends on the definition of knowledge, we must first weaken our definition of knowledge. We introduce a modal operator K_i^d which is intended to mean that "i is justified in believing ...". Our approach will be to define a system of Grove spheres on the set $\mathcal{H}$ [49].

Definition 13 *Let $\mathcal{H}$ be a set of global histories. A* **system of spheres** *on $\mathcal{H}$ is a set $\mathbb{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots\}$ where for each $i \geq 1$, $\mathcal{S}_i \subseteq \mathcal{S}_{i+1} \subseteq \mathcal{H}$, and $\cup_{i=1}^{\infty} \mathcal{S}_i = \mathcal{H}$.*

The intuition is that if $i < j$, then the histories in $\mathcal{S}_i$ are "more plausible" than those in $\mathcal{S}_j - S_i$. There are two issues that will be important. The first is: *Given a finite global history H, which histories are the most plausible given that the situation has evolved according to H?* Denote this set of histories by $\mathcal{D}(H)$. Of course we want $\mathcal{D}(H) \subseteq \mathcal{F}(H)$ (the set of global histories extending H). In order to formally define $\mathcal{D}$, we define an index function I for a system of spheres $\mathcal{S}$. Given a finite global history H, $I(H) = \mu i.(\exists H', H'_i = H \text{ and } H' \in \mathcal{S}_i)$, i.e., $I(H)$ is the smallest index of a sphere

containing an infinite extension of H. Then given a finite global history H,

$$\mathcal{D}(H) = \mathcal{S}_{I(H)} \cap \mathcal{F}(H)$$

That is, $\mathcal{D}(H)$ is the set of the most plausible histories extending H. The second notion is the set of global histories that a particular agent considers most plausible given the events he has seen. Formally, let $i \in \mathcal{A}$ be an agent and suppose that h is a local history for agent i. Then define the i-index function $I_i(h) = \mu j.(\exists H \in \mathcal{S}_j, \quad \lambda_i(H_t) = h)$, where $t = \mathsf{len}(h)$ (the length of h).

So, $I_i(h)$ is the least index of a sphere containing a history in $\lambda^{-1}(h)$. We can then define the set of histories that i considers plausible, given the events that i has seen. Denote this set $\mathcal{D}_i(h)$ and define it as follows

$$\mathcal{D}_i(h) = \{H' \mid \lambda_i(H'_t) = h\} \cap \mathcal{S}_{I_i(h)}$$

and here t is the length of all finite histories, local and global, mentioned.

We can now define the notion of *weak knowledge*. We say that agent i *justifiably believes* ϕ at H, t, denoted $K_i^d\phi$, if ϕ is true in all i-plausible histories. Formally,

$$H, t \models K_i^d\phi \text{ iff for all } H', H'_t \in \mathcal{D}_i(\lambda_i(H_t)), \; H', t \models \phi$$

or putting it in terms of the local history h

$$h \models K_i^d\phi \text{ iff for all } H', H'_t \in \mathcal{D}_i(h),\ H', t \models \phi$$

Of course $K_i\phi$ semantically entails $K_i^d\phi$. In general K_i^d does not satisfy the veridicality axiom (the truth of $K_i^d\phi$ does not necessarily imply that ϕ is true). But it is easy to check that both positive and negative introspection hold. That is

Lemma 3.4.1 *K_i^d satisfies both positive and negative introspection. That is the following schemes are valid.*

1. $K_i^d\phi \rightarrow K_i^dK_i^d\phi$

2. $\neg K_i^d\phi \rightarrow K_i^d\neg K_i^d\phi$

Proof Suppose that $H, t \models K_i^d\phi$. Then for any H' with $H'_t \in \mathcal{D}_i(\lambda_i(H_t))$, $H', t \models \phi$. Let H'' and H''' be arbitrary histories such that $H''_t \in \mathcal{D}_i(\lambda_i(H_t))$ and $H'''_t \in \mathcal{D}_i(\lambda_i(H''_t))$. Since $H'' \in \mathcal{D}(\lambda(H_t))$, $\lambda(H''_t) = \lambda_i(H_t)$ and since $H'''_t \in \mathcal{D}(\lambda(H''_t))$, $\lambda_i(H'''_t) = \lambda_i(H''_t)$. Therefore, $\lambda_i(H'''_t) = \lambda_i(H_t)$ and hence since $I_i(\lambda(H_t)) = I_i(\lambda(H''_t))$, we have $\mathcal{S}_{I_i(\lambda(H_t))} = \mathcal{S}_{I_i(\lambda(H''_t))}$. Therefore, $\mathcal{D}_i(\lambda_i(H_t)) = \mathcal{D}_i(\lambda_i(H''_t))$. Hence, since $H'''_t \in \mathcal{D}_i(\lambda_i(H_t))$, we have $H''', t \models \phi$. Therefore, $H'', t \models K_i^d\phi$ and since H'' was arbitrary, $H, t \models K_i^dK_i^d\phi$. The proof of 2 is similar. □

Thus the logic of the operator K_i^d is $\mathbf{KD45}_n$ rather than $\mathbf{S5}_n$, but we do *act* as if it were $\mathbf{S5}_n$. We act on the advice of the short story writer Damon Runyon, "The race is not always to the swift, nor the battle to the strong, but that is the way to bet." In short, if a is the best action given ϕ, and $K_d^i(\phi)$ holds, then do a.

3.4.1 Default Obligations

The obligation defined in Definition 11 is an *absolute obligation for agent i* in the sense that the obligation remains until a required action is performed by the agent. No amount of information, however surprising, can remove the obligation. But this is not the case for Jill's obligation in example 4. Jill loses the obligation to administer drug d upon learning from nurse Rebecca that the patient is allergic to drug d. In this example, Jill not only gained the obligation to administer drug d' upon learning some surprising information, but also lost an obligation to administer d. Thus Jill's obligation to administer drug d was a *default obligation*, as an absolute obligation could not be lost.

The machinery we developed in this section can be used to formalize this notion. We say that an agent i has a default obligation to perform action a, provided all maximal histories that the agent considers plausible are ones in which a is performed. Formally

Definition 14 *Agent i has a* default obligation *to perform action a at global history H and time t iff a is an action which i (only) can perform, and i* justifiably believes *that it is good to perform a. Putting this in terms of the agent's local history* $h = \lambda_i(H_t)$*, all*

maximal extensions of any $H_t' \in \mathcal{D}_i(h)$ *belong to the range of the action* a.

Clearly, if agent i is obliged to perform action a, then agent i also has a default obligation to perform action a. There are three notions which are important for this chapter. Let H be a global history, $t \in \mathbb{N}$ and a an action.

1. a is a *good to be performed* at H, t iff every maximal extension of H_t is in the range of a, i.e., $\mathcal{G}(H_t) \subseteq a(H_t)$

2. a is a *knowledge based obligation* at H, t iff a satisfies Definition 11

3. a is a *knowledge based default obligation* at H, t iff a satisfies Definition 14.

If a is a good action, then a *ought* to be done, but the agent in question might not have any reason to believe that a ought to be done. This framework can now be used to understand the following quite well-known example.

The Kitty Genovese Murder

"Along a serene, tree-lined street in the Kew Gardens section of Queens, New York City, Catherine Genovese began the last walk of her life in the early morning hours of March 13, 1964.....As she locked her car door, she took notice of a figure in the darkness walking towards her. She became immediately concerned as soon as the stranger began to follow her.

'As she got of the car she saw me and ran,' the man told the court later, 'I ran after her and I had a knife in my hand.... I could run much faster than she could, and I jumped on her back and stabbed her several times,' the man later told the cops."

Many neighbours saw what was happening, but no one called the police.

"Mr. Koshkin wanted to call the police but Mrs. Koshkin thought otherwise. 'I didn't let him,' she later said to the press, 'I told him there must have been 30 calls already.' "

"When the cops finished polling the immediate neighbourhood, they discovered at least 38 people who had heard or observed some part of the fatal assault on Kitty Genovese."[3]

Some 35 minutes passed between Kitty Genovese being attacked and someone calling the police. Why?

Analysis: The people who saw Kitty being killed did not have default knowledge that they had the obligation to help her. They all knew that the good histories were ones in which *someone* called the police, but not all these histories were ones where *they themselves* were the caller – someone else could be the caller. Compare this to a situation in a waiting room where a child's mother goes to the bathroom and her daughter starts to cry. Again there is no one who has a default obligation to comfort the child, but

[3]This quote is from the article 'A cry in the night: the Kitty Genovese murder', by a police detective, Mark Gado, and appears on the web in *Court TV's Crime Library*.

typically, if there is a woman in that waiting room, she will *see* that no one else is taking care of the child and assume responsibility. Unlike the Geneovese case, there will be a common knowledge, *until the child is comforted*, that the child is not being comforted. Well designed social software, (a notion defined originally in [76]) will address such issues.

As we saw earlier there is not only knowledge but justifiable belief and the justifiable belief of what are the best histories will depend on what one thinks the histories are. A man at the beach alone who sees a boy drowning will surely do something. There *may* be someone watching from a distance who might be a better swimmer than he himself is. But his default is that he is the only one who knows, is present, and therefore has the obligation to help. If on the other hand he is among 50 people at the beach, then he no longer has default knowledge of his obligation. There might well be other people on the beach who are better swimmers than he is, and perhaps among them are the boy's companions. Mrs. Koshkin's admonition to her husband amounted to her saying to him, "You do not have the default obligation."

3.5 Putting Everything Together

We have developed quite a bit of machinery in this chapter, and so at this point it is worthwhile to summarize our discussion so far. We begin by extending the language

$\mathcal{L}_n^{KT}$ to $\mathcal{L}_n^{KBO}$. Formulas in $\mathcal{L}_n^{KTO}$ have the following syntactic form:

$$\phi := p \mid \neg\phi \mid \phi \wedge \phi \mid \bigcirc\phi \mid \phi\, U\psi \mid K_i\phi \mid [a]\phi \mid G(a)$$

where $p \in \mathsf{At}$ and $a \in \mathsf{Act}$. We define the standard boolean operators, L_i and the temporal operators F and G as usual (see Chapter 2). Define $\langle a\rangle\phi$ to be $\neg[a]\neg\phi$. Let $\mathcal{L}_n^{K^dTO}$ be the language which is just like $\mathcal{L}_n^{KTO}$ but replace each K_i modality with K_i^d. We now give the intended interpretation of some of the formulas in $\mathcal{L}_n^{KTO}$ ($\mathcal{L}_n^{K^dTO}$).

- $G(a)$: "action a is good"', or "a is a non-informational obligation"
- $\langle a\rangle\top$: "action a can be performed'
- $K_i\langle a_i\rangle\top$: "agent i knows that she can perform action a_i"
- $K_iG(a_i)$: "agent i knows that action a_i is good", i.e., "i is obliged to perform a_i". Note that we will have $H, t \models K_i(G(a_i))$ just in case a_i is a knowledge based obligation for agent i at H_t (Definition 11).
- $K_i^d\phi$: "agent i weakly knows ϕ"
- $K_i^dG(a_i)$: "agent i has a default obligation to perform a_i". Note that we will have $H, t \models K_i^d(G(a_i))$ just in case a_i is a default obligation for agent i at H_t (Definition 14).

We will now repeat the four examples from the introduction and show how to formalize each example in the language $\mathcal{L}_n^{KTO}$ ($\mathcal{L}_n^{K^dTO}$). Let $\mathcal{A} = \{j, s, a, r\}$ be the set of agents (with the obvious interpretations) and suppose that $\mathsf{Act} = \{v, t, m\}$ are the set of actions (the interpretations will be given below).

Example 1: Jill is a physician whose neighbour is ill. Jill does not know and has not been informed. Jill has no obligation (as yet) to treat the neighbour. Formally, $\neg K_j G(r)$, where r is the action of treating the neighbor (which only Jill can perform).

Example 2: Jill is a physician whose neighbour Sam is ill. The neighbour's daughter Ann comes to Jill's house and tells her. Now Jill does have an obligation to treat Sam, or perhaps call in an ambulance or a specialist. Formally, $K_j(G(r))$ is true. The interesting thing about this example is that this formula becomes true, because at the previous moment, $K_a(G(m))$ is true, where m stands for the action of telling Jill about Sam's illness (which only Ann can perform) *and* that Ann actually did send the message m. We discuss this example in more detail below, in particular we are interested in capturing Ann's reasoning which allows her to conclude that m is an obligatory action.

Example 3: Mary is a patient in St. Gibson's hospital. Mary is having a heart attack. The caveat which applied in example 1. does not apply here. The hospital has

an obligation to *be aware* of Mary's condition at all times and to provide emergency treatment as appropriate. The issue here falls outside of the scope of our discussion thus far. What is important for this example is that the hospital has an obligation to ensure that procedures are set up to guarantee that at each moment $K_j(G(r))$ (here r means treat the next patient). What complicates matters from the hospital's point of view is that the hospital cannot necessarily assume that all agents are using the same value function. Hence, the task of the hospital is to set up social procedures plus a system of rewards and punishments so that the agent's behave as if they are using the same value function. We briefly touch on these issues in Section 3.7.

Example 4: Jill has a patient with a certain condition C who is in the St. Gibson's hospital mentioned above. There are two drugs d and d' which can be used for C, but d has a better track record. Jill is about to inject the patient with d, but unknown to Jill, the patient is allergic to d and for this patient d' should be used. Nurse Rebecca is aware of the patient's allergy and also that Jill is about to administer d. It is then Rebecca's obligation to inform Jill and to suggest that drug d' be used in this case. Let δ stand for the action of giving drug d to the patient, similarly for δ' and d'. Formally, Jill has the default obligation to give the patient drug d $(K_j^d(G(\delta)))$. However, since Rebecca knows that Jill has this default obligation $(K_r K_j^d(G(\delta)))$, Rebecca has an obligation to inform Jill about the drug $(K_r(G(m_d)))$ where m_d means tell Jill about the allergy to

drug d). Of course, we can replace each of Rebecca's knowledge operators with a weak knowledge operator.

Before turning to the semantics, we point out an issue relevant to our analysis. If $\langle a_i \rangle \top$ is true at some finite history H, then this represents that agent i *can* perform action a_i at history H. But this does not mean that agent i actually *does* perform actions a_i. In fact, our formal language does not have any machinery to express such a statement. Thus a question arises as to whether or not an agent will actually perform a given that the agent knows that it is good. This is important for Example 2 as we need not only that Ann knows that she should send a message to Jill, but also that Ann actually does send the message. This is relevant to our discussion because we are assuming that the agents share a utility function. Thus if an agent knows that a is good to perform, then the agent knows that it is in its own best interest to perfrom a. One is tempted to conclude, that *of course* the agent will perfrom a in this case. Davidson considers these and related issues in [31]. These issues are relevant to the question of which protocols are considered plausible which is not central to the discussion at hand. We now turn to the semantics.

Definition 15 *Let $\mathcal{F}_K$ be a history based frame. A* **knowledge based obligation model** *based on $\mathcal{F}_K$ is a structure $\mathcal{M}_O = \langle \mathcal{H}, \{E_i\}_{i \in \mathcal{A}}, \{\lambda_i\}_{i \in \mathcal{A}}, \{\mathsf{Act}_i\}_{i \in \mathcal{A}}, \mathsf{val}, V \rangle$ where*

- *$\mathcal{H}$ is a closed protocol satisfying the single agent property*

- *The set of actions for each agent are pairwise disjoint and for each* $i \in \mathcal{A}$, $\mathsf{Act}_i \subseteq E_i$
- val *is a value function (Definition 10)*
- V *is a valuation function*

Truth in the model is defined as usual. We only give the definition of the new formulas:

- $H, t \models [a]\phi$ iff for all $H' \in a(H_t)$, $H', t+1 \models \phi$
- $H, t \models G(a)$ iff $\mathcal{G}(H_t) \subseteq a(H_t)$

Note that $G(a) \rightarrow \langle a \rangle \top$ will be valid in any knowledge based obligation model. This follows since the conditions on the val function implies that for any finite history H, $\mathcal{G}(H)$ is non-empty.

A **default knowledge based obligation model** extends a knowledge based obligation model with a system of spheres. That is a default knowledge based obligation model is a structure $\mathcal{M}_{O^d} = \langle \mathcal{H}, \{E_i\}_{i \in \mathcal{A}}, \{\lambda_i\}_{i \in \mathcal{A}}, \{\mathsf{Act}_i\}_{i \in \mathcal{A}}, \mathsf{val}, \mathbb{S}_{\mathcal{H}}, V \rangle$, where each component is as above and $\mathbb{S}_{\mathcal{H}}$ is a system of spheres on $\mathcal{H}$ (see Definition 13).

3.6 Formalizing the Examples

In this section, our goal is to show that the formal machinery we have developed in this chapter can be used to capture our intuitions about each of the examples from the introduction. We will only discuss examples 1, 2 and 4. As stated in the previous

section, example 3 deals with different issues, and so we will not discuss it in this section. Section 3.7 discusses some issues relevant to example 3. More specifically, our task is to construct a knowledge based obligation model in which the formulas from the previous section have their requisite truth values.

We begin by constructing a protocol $\mathcal{H}$. There are four events, v, m, r, c where v stands for Sam vomiting, m stands for Ann telling Jill, r stands for Jill treating (or offering to treat) Sam and c is a clock tick which, unlike the other three, may occur more than once. Our global histories will consist of sequences in which events occur infinitely often, but v, m, t occur at most once. Moreover, since we assume Ann is truthful, m never occurs without v occurring first. Let $\mathcal{H}$ be the set of all such histories (closed under finite prefixes).

To be more precise, let $\mathcal{A} = \{j, s, a, r\}$ (with the obvious interpretation); and $\mathsf{Act}_j = \{r\}$, $\mathsf{Act}_a = \{m\}$, and $\mathsf{Act}_s = \{v\}$. Assume that the event v is observed by Sam and Ann, m by Ann and Jill, and r and c, let us say, by all three. That is, $E_j = \{r, m, c\}$, $E_a = \{r, v, m, c\}$, $E_s = \{r, v, c\}$. Then $\mathcal{H} \subseteq E^\omega = (E_j \cup E_a \cup E_s)^\omega$ is the set of global histories (closed under finite prefixes) described above. It is easy to see that by construction $\mathcal{H}$ satisfies the single agent property. Furthermore, $\mathcal{H}$ is easily seen to be closed (this follows since we start by creating a set of infinite global histories, then construct $\mathcal{H}$ by closing under the finite prefix function).

The next set of assumptions concern the values of each global history. In those finite

global histories in which v has occurred but not yet r, the best continuations are those in which r now occurs. And if v has not yet occurred then r (in the form of an offer to treat) may occur, but makes the history worse as the doctor is embarrassed by offering to treat a healthy man. Thus we stipulate that all histories in which neither v nor r occurs have value 2, those in which r occurs without v have value 1 as do those in which v is followed by r. Finally those histories in which v occurs but not r have value 0 as they are the worst. Let val be a value function that assigns the global histories these values and let $\mathcal{M}_0$ be the knowledge based obligation model we have just sketched (actually it is only a frame since we have not specified the truth values of the propositional variables).

It is convenient to introduce a propositional variable that can be used to describe properties of the histories (for example whether or not Sam is sick). Let sick be a propositional variable that is true at any finite history in which v has occurred without r. It is worth pointing out that sick is a description of events that have or have not taken place, not a description of how Sam feels. Otherwise, we would be assuming that Jill's treatment *always* cures Sam.

Suppose now that an agent's local history is h and that the agent acquires some knowledge. In that case, the set of global histories H such that $\lambda_i(H_t) = h$ will *decrease*, and universal quantified formulas over all such histories will be more likely to become true. Thus before Jill was told of Sam's illness, the set of global histories compatible with her own local one included many where Sam was not ill. Receiving the information,

however, deletes them, and in all global histories still compatible with her knowledge, she must act to help Sam. Similarly, in example 2 Ann had an obligation to inform Jill, for before she tells Jill, in many of *Jill's* local histories compatible with Ann's, and in some global histories compatible with these latter, Ann's father is not ill and Jill cannot act. By informing Jill, Ann extends Jill's local history, and creates an obligation for Jill. Moreover, assuming that Ann knows that Jill does what she ought to, Ann herself has the obligation to inform Jill.

We first consider examples 1 and 2 from Jill's point of view. In a history in which v has occurred but not m, from Jill's point of view there are global histories in which v has not occurred which are compatible with her own local history. So she cannot know that it is good to treat Sam, although it is. She is not yet obligated to treat Sam. Once m occurs, she knows that v must have occurred, it is good to treat, and she knows it. So she is obligated. More formally, we can show that $(K_j\mathsf{sick} \wedge \langle t \rangle \top) \rightarrow K_j G(r)$ is valid in $\mathcal{M}_O$. Furthermore, if we assume that Jill is "ethical" (i.e., her utility function matches the global value function), then we can conclude that if $K_j(G(r))$, then Jill will in fact choose to treat Sam. Finally, the obligation arises to treat same *only* because Jill knows that Sam is ill, i.e., $\neg K_j\mathsf{sick} \rightarrow \neg K_j(G(r))$. The following observation makes our claim more precise.

Observation 3.6.1 *Let $\mathcal{M}_O$ be the knowledge based obligation model sketched above. Then the following formulas are valid in $\mathcal{M}_O$*

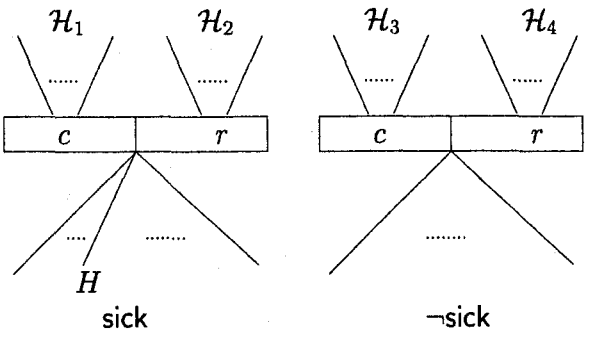
1. $(K_j\mathsf{sick} \wedge \langle t \rangle \top) \rightarrow K_j G(r)$

2. $\neg K_j\mathsf{sick} \rightarrow \neg K_j(G(r))$

First consider formula 2. This represents the situation in example 1. That is Jill does not know that Sam is ill, so she does not have the obligation to treat Sam. Let H be an arbitrary global history and $t \in \mathbb{N}$ an arbitrary moment such that $H, t \models K_j\mathsf{sick}$. Now, by the construction of $\mathcal{H}$, for any H' such that $H_t \sim_i H'_t$, m does not occur in H'_t. This follows since we assume Jill is aware of m and m only occurs in histories in which v has occurred. Furthermore, if r cannot be performed at H_t, then trivially $H, t \models \neg K_j(G(r))$ (since in this case $H, t \not\models G(r)$). Finally, it is not hard to see that in the construction of $\mathcal{H}$ we have assumed that Jill can *choose* whether or not to perform action r. As such we can assume the following. There are four subsets of global histories $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ and $\mathcal{H}_4$ such that

- $\mathcal{H}_1 = \{H' \mid H'_t c \preceq H' \text{ and } H_t \sim_j H'_t \text{ and } H', t \models \mathsf{sick}\}$
- $\mathcal{H}_2 = \{H' \mid H'_t t \preceq H' \text{ and } H_t \sim_j H'_t \text{ and } H', t \models \mathsf{sick}\}$
- $\mathcal{H}_3 = \{H' \mid H'_t c \preceq H' \text{ and } H_t \sim_j H'_t \text{ and } H', t \models \neg\mathsf{sick}\}$
- $\mathcal{H}_4 = \{H' \mid H'_t t \preceq H' \text{ and } H_t \sim_j H'_t \text{ and } H', t \models \neg\mathsf{sick}\}$

For simplicity, assume that $H \in \mathcal{H}_1$. This situation can be pictured as follows:



The above picture shows all the global histories that are equivalent from Jill's point of view at time t. These global histories can be grouped into two disjoint sets: the ones in which v has occurred and the ones in which v has not occurred. Each of the sets can be further divided into ones in which Jill performs action r and those in which Jill performs the (in)action c. Now, the definition of the value function implies that $\max(\mathsf{val}[\mathcal{H}_1]) = 1$, $\max(\mathsf{val}[\mathcal{H}_2]) = 2$, $\max(\mathsf{val}[\mathcal{H}_3]) = 2$ and $\max(\mathsf{val}[\mathcal{H}4]) = 1$. In other words, if the neighbor is sick then it is strictly better to treat the neighbor than to not treat the neighbor; however if the neighbor is not sick, then treating the neighbor for an illness he does not have is worse than not treating the neighbor. Let $H' \in \mathcal{H}_3$ be a history with maximal value (with respect to the histories in $\mathcal{H}_3$). Then since $H' \notin t(H'_t)$, we have $\mathcal{G}(H'_t) \not\subseteq t(H'_t)$ and so $H', t \not\models G(r)$. Therefore, since $H_t \sim_j H'_t$, $H, t \not\models K_j G(r)$. Thus Jill is not obliged to perform action r. Essentially, we are comparing the functions r and c on a domain D of histories compatible with Jill's local history. On this domain r and c are not comparable, neither dominates the other.

For the first formula, suppose that Ann informs Jill that her father is sick (as in

example 2). Actually all that is needed to be assumed is that Jill can rule out the right half of the above picture, i.e., all the histories in which v has not occurred (it does not matter *how* she came upon this information). However, we will focus on example 2. The message from Ann changes Jill's local view so that the sets of histories $\mathcal{H}_3$ and $\mathcal{H}_4$ are no longer possible for her. Jill's local view restricts the set of possible global histories to $\mathcal{H}_1$ and $\mathcal{H}_2$. And so, Jill *is* obliged to perform action a, since for any history on the new domain of histories compatible with Jill's updated local view, r is strictly better than the (in)action c. Formally, if we assume that event m has occurred, then Jill rules out all global histories in which v has not occurred. Notice that for Jill this effect is achieved by assuming that in $\mathcal{H}$ there are no global histories that contain m alone. This amounts to assuming that Ann is honest, i.e., if she sends a message about her father's illness it is only because v has occurred (and this is common knowledge). Thus if H is a global history in which m has occurred (at time $t-1$), then for all H' with $H_t \sim H'_t$, since v must have occurred in H', $H', t \models \mathsf{sick}$ and so $H, t \models K_j\mathsf{sick}$. Now, we have that for each H' such that $H'_t \sim_j H_t$, $\mathcal{G}(H_t) = \{H' \mid H'_t t \preceq H'\} = t(H_t)$ and so $H', t \models G(r)$. Hence $H, t \models K_j G(r)$. Thus Jill has the (knowledge based) obligation to treat Sam.

We know consider the situation from Ann's point of view. Suppose again that v has occurred but not m yet. Then according to Ann, Jill's local history is compatible with v not having occurred and in fact we will have that $K_a(\neg K_j(\mathsf{sick}))$ (Ann knows that Jill does not know about the vomiting). This formula will be true provided Ann *knows*

Jill's local history. Of course it is unrealistic that Ann knows all of Jill's local events, but it is enough for Ann to know enough about Jill's histories so that Ann knows that probably Jill considers it possible that Sam has not vomited, i.e., $K_a^d(\neg K_j(\mathsf{sick}))$ (or perhaps $K_a^d(\neg K_j^d(\mathsf{sick}))$).

Since the vomiting *has* happened, all good histories now are those in which Sam has been treated, and those are included in the ones in which Ann has told Jill. So Ann ought to inform Jill about v, i.e. cause the event m. Formally, we have for any infinite global history H and time $t \in \mathbb{N}$, $H, t \models K_a\mathsf{sick} \wedge \langle m\rangle\top \rightarrow K_a(G(m))$. The proof of this fact is analogous to the argument concerning Jill. Let H be a fixed global history and $t \in \mathbb{N}$. The idea is that in our model, the maximal histories that extend H'_t, where $H_t \sim_a H'_t$ all contain the event m. In fact, more can be said about this situation. The analysis so far does not explain *why* Ann concludes that she should send the message m to Jill. We will discuss this in more detail in the next section, but for now we show that the following formula is valid in our model: $[m]K_jG(r)$. Essentially, the reason is that we only consider histories that if they contain m then they must contain v, i.e., Ann is truthful (and this fact is common knowledge). So if F is an arbitrary history and $t \in \mathbb{N}$, then for each globally history $F' \in a(F_t)$, F' is a history in which both m and v have taken place. Then using the above argument, $F', t \models K_jG(r)$. Hence $F, t \models [m]K_jG(r)$. Since it is true for arbitrary global histories, then it will certainly be true at histories which are equivalent to H_t according to Ann. Hence, $H, t \models K_a[m]K_jG(r)$.

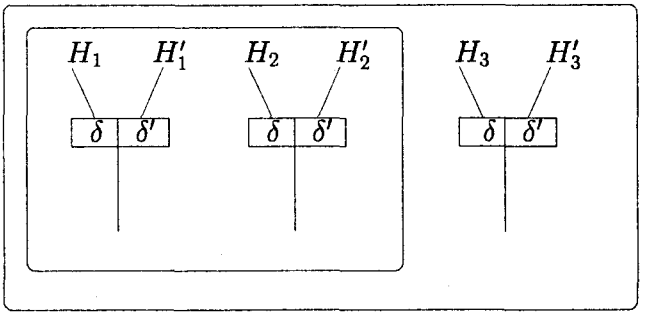
In a more complex scenario, with other agents, it could of course be that someone other than Ann had informed Jill of Sam's illness, but that Ann does not know this. We would say that Ann still has a default obligation to inform Jill, and this can easily be expressed in our language. Also note that in our scenario, once the obligation to treat arises, it remains until treatment has taken place.

The case of the nurse Rebecca is a bit more tricky. The reason is that acquiring knowledge may create an obligation as we saw before, but it cannot erase an absolute one. The existence of an obligation is a universally quantified formula whose truth value can only go from *false* to *true* as the domain shrinks. Thus if Jill had an absolute obligation to administer drug d before being informed by Rebecca of Mary's allergy, then she would still have it. How, then do we represent the fact that on learning of the allergy she *acquires* the obligation to administer d' but *loses* the obligation to administer d?

As discussed in Section 2.3, to deal with this case we will use the notion of a default history. Those histories in which patients do not have this allergy may be regarded as the usual kind and those in which they do are unusual. Typically, obligations are evaluated in terms of histories of the usual kind and when we say "good" history, we mean a good history of the usual kind. Learning about the allergy deletes these usual histories, and then the action contemplated is re-evaluated in terms of the unusual variety. Thus d is better than d' when we consider the usual sort of history, but the opposite happens

when we consider the unusual variety.

The following picture illustrates the above discussion. Suppose that δ is the action 'give drug d to Mary' and δ' is the action 'give drug d' to Mary'. Suppose that according to Jill's information, each of the histories H_i is indistinguishable from H_j for $i, j = 1, \ldots, 3$ and similarly for the H_i', H_j'. Also that $\mathsf{val}(H_i) > \mathsf{val}(H_i')$ for $i = 1, 2$, but $\mathsf{val}(H_3') > \mathsf{val}(H_3)$. In this case Jill is not absolutely obliged to perform δ since $\mathsf{val}(H_3') > \mathsf{val}(H_3)$. However, if the histories H_3 and H_3' are only *remotely* possible, then Jill is obliged to perform action δ, i.e., administer drug d. In figure 2, the histories inside the innermost rectangle are the "usual" histories. Once Rebecca informs Jill about Mary's allergy, the histories inside the rectangle are no longer possible; and so Jill is now obliged to perform action δ' and no longer obliged to perform δ.



3.6.1 Common Knowledge of Ethicality

Note that many of the arguments in the previous section boiled down to assumptions about which strings belong to the protocol under consideration. As such, the analysis

may have appeared ad hoc. In this section we argue that the assumptions we made about the protocol in the previous section were not ad hoc, but rather follow from a general principle. We call this assumption *Common Knowledge of Ethicality.* Before discussing this principle, we go into some more details about Ann's reasoning.

At this point it is convenient to introduce some propositional variables which will make the discussion easier to follow. Recall that sick is a propositional variable which is true at all finite histories in which v has occurred. Similarly, define treat to be true at exactly those histories in which r has occurred and msg to be true at those histories in which m has occurred. We argued in the previous section that Ann has the (knowledge based) obligation to tell Jill about her father's illness. Clearly, Ann will not be under any obligation to tell Jill that her father is ill, if Ann (weakly) knows that Jill would not treat her father even if she knew of his illness. Thus, to carry out a deduction we will need to assume

$$K_a(K_j\mathsf{sick} \leftrightarrow \bigcirc\mathsf{treat})$$

This says that Ann knows that Jill will treat (at next moment) iff she knows that Sam is ill. A similar assumption is needed to derive that Jill has an obligation to treat Sam. Obviously, if Jill has a good reason to believe that Ann always lies about her father being ill, then she is under no obligation to treat Sam. In other words the following formula must be true

$$K_j(\mathsf{msg} \leftrightarrow \mathsf{sick})$$

This formula says that Jill knows that a message is sent iff Ann's father is ill.

These formulas can all be derived for one common assumption which we call *Common Knowledge of Ethicality.* Analogous to the common knowledge of rationality in the game theory literature, this assumption assumes that each of the agents are ethical, everyone knows that they are ethical, everyone knows that everyone knows that they are ethical, and so on. Here of course, "ethical" simply means that the agent's personal utility function matches the social value function.

Although this assumption of common knowledge of ethicality is needed in order to fully understand our examples, we do not need to include it explicitly as it is tacitly included in the set $\mathcal{H}$ which are considering. E.g. we simply *leave out* histories in which Jill knows about Sam's illness but fails to treat him. A more ambitious analysis would start with a *larger* $\mathcal{H}'$ and then use the common knowledge of ethicality to *cut down* to the sort of $\mathcal{H}$ we are using.

3.7 Programming the Agents

In this section, we make a quick digression into some of the issues relevant for the analysis of example 3. Namely, we will consider the following question. Given a set of histories and values assigned to each history, we can ask, "Is it possible to program the agents in such a way that *if the agents do what the know they ought to do*, then one of the best histories is produced?

We first must decide how much computational power we will ascribe to the agents. Assuming that agents have perfect recall requires that they have unbounded memory, and we will need to model them as Turing machines; however, we may want to assume that the agents only need to remember a bounded amount of information. In this case we will assume that the agents are finite automata. We will sketch how to design finite automata which will generate an interpreted system (see [34] Chapter 5 and Section 2.2.1 for a definition) in which we can show that the agents will have the required knowledge based obligation. See [63] for a discussion of adding the notion of obligation to interpreted systems.

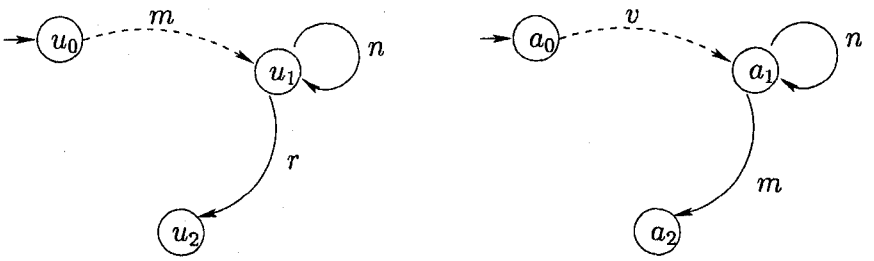
Consider the example where Ann is obliged to inform Jill about her father's vomiting which induces Jill to have the obligation to treat Sam (Ann's father). We assume $E = \{v, m, t, c\}$, where v stands for vomiting, m for Ann telling Jill about her father's illness, r for Jill treating Sam and c for a clock tick. Thus histories are strings over E. For the conditions placed on these strings, refer to Section 3.6.

Since in this example, Sam has no control over whether or not he vomits, we only consider Jill and Ann. We can ascribe the following finite automata to Jill and Ann. For Jill, suppose that the input alphabet is $\Sigma_U = \{m, t\}$, the states are $Q_U = \{u_0, u_1, u_2\}$ with u_0 being the start state. As for the transitions, we need to consider two types of transitions. The first is a transition induced by an action of another agent. For example, when Jill is in state u_0, and she "sees" an m, she moves to state u_1. Since m is not an

action that Jill can perform, we think of this transition as being forced or caused by another agent (Ann in this case). Now once Jill is in state u_1, it is her turn to act. She can move to state u_2 by performing action r or simply stay in state u_1 by doing nothing. But in any case *knowing* of v corresponds to being in state u_1.

Ann's automaton will be similar. Let $\Sigma_A = \{v, m\}$ and $Q_A = \{a_0, a_1, a_2\}$. Ann's initial state is a_0, when her father vomits she transitions from a_0 to a_1. While in a_1 she can choose to do nothing or perform action m to move to state a_2. But she *knows* of v as she is in state a_1.

Figure 3 depicts the above finite automata. The dashed lines represent transitions induced by other agents or the environment, and the solid line represents the choices that each agent can make.



Define values as follows. All histories in which neither v nor r occurs have value 2, Those histories in which v occurs but not r have value 0 as they are the worst. Those histories in which v is followed by r are assigned as follows. Let H be a history in which v is followed by r, $\mathsf{val}(H) = \frac{1}{N+1} + \frac{1}{M+1}$, where N is the number of clock ticks between the

occurrence of v and m, and M is the number of clock ticks between the occurrence of m and r. Those in which r occurs without v have value 1 as do those in which v is followed by r. This valuation not only means that both Ann and Jill have to act, but that they should act speedily, for any delay leads to histories with lower values. Notice that since we are assuming that the agents are finite automata, they will not *know* the value of the generated histories. Thus the desired effect of assigning these values to the histories may be lost on the agents. We can achieve a similar effect by assuming that when Jill is in state u_1 and Ann is in state a_1, then the agents feel a sense of "urgency" to leave this state, i.e. perform r and m respectively. This issue will be discussed in more detail in future work.

3.8 Conclusions

A central issue when designing a social procedure is how to ensure that the agents will perform the required actions. One may suspect that the situation is trivial if we can assume that the agent's all share the same utility function. The examples discussed in the introduction show that this is not the case. The information state of the agent is crucial when determining whether or not the agent is obliged to perform an action. This chapter provides a formal framework for reasoning about agents in social situations that are assumed to share a utility function. We start with the intuition that agents should not be faulted for not performing actions that they do not know about, and develop a

formal language and semantics for reasoning about obligations, actions and knowledge in a multi-agent setting. The main contribution of this chapter is conceptual, and indeed a number of technical questions remain. Nonetheless, we have showed that the formal machinery developed in this chapter can be used to formalize the four illustrative examples from the introduction and so provides a powerful framework for reasoning about social software.

The main technical issue which remains is a sound and complete axiomatization. Of course, we should begin with the axiomatization $\mathbf{S5}_n^U$ from Chapter 2 for the knowledge and temporal modalities and add the required axioms that correspond to the agent's reasoning capabilities (in this case perfect recall and common knowledge of the global clock). Finding the right axioms that connect our obligation formulas and our knowledge formulas requires making the common knowledge of ethicality more explicit. One obvious solution is to introduce a common knowledge operator into our language together with the standard axiomatization. However, as discussed in Chapter 2 this greatly increases the complexity of the validity problem and even in some cases makes the validity problem Π_1^1 complete. In particular, if the agents are assumed to have perfect recall and have access to the global clock, then if we add a common knowledge operator to our language (with the standard interpretation), no recursive axiomatization is possible. Thus we need to find a way to bring in the assumption of common knowledge of ethicality without explicitly introducing a common knowledge operator. This will be left

for further research.

Chapter 4

Communication Graphs

The previous chapter focused on defining a notion of obligation that *depends* on an agent's state of information. That is, we were interested in the effect of an agent's state of information on its choice of action. So, as the agent's information changes so do its obligations. In this chapter, we shift our focus to *how* agents can change their states of information. In particular, we are interested in representing the dynamic effects of communication on each agent's state of information. This dynamic shift in the study of formal models of knowledge and beliefs has recently gained a lot of attention.

The study of *Dynamic Epistemic Logic* attempts to combine ideas from dynamic logics of actions and epistemic logic. The main idea is to start with a formal model that represents the uncertainty of an agent in a social situation. Then we can define an 'epistemic update' operation that represents the effect of a communicatory action, such

as a public announcement, on the original model. For example, publicly announcing a true formula ϕ, shifts from the current model to a submodel in which ϕ is true at each state. Starting with [89] and more recently [7, 61, 106, 44, 101], logical systems have been developed with the intent to capture the dynamics of information in a social situation. Chapter 4 of Kooi's dissertation [61] an the forthcoming article [62] surveys of the current state of affairs.

In this chapter, which is based on joint work with Rohit Parikh [66], we develop a multi-agent epistemic logic with a communication modality. Agents are assumed to have some private information at the outset, but may refine their information by acquiring information possessed by other agents, possibly via yet other agents. Each agent's information is initially represented by a partition over a set of possible states. Agents are assumed to be connected by a *communication graph.* In the communication graph, an edge from agent i to agent j means that agent i can directly receive information from agent j. Agent i can then refine its information by learning information that j has, including information acquired by j from another agent, k. We introduce a multi-agent modal logic with knowledge modalities and a modality representing communication among agents.

The semantics presented in this section is a combination of the history based frames discussed in Chapter 2 and subset models introduced by Parikh and Moss in [64]. The next section is an introduction to subset models and *Topologic.* Section 4.2 introduces

the logic of communication graphs and shows that the satisfiability problem is decidable. Finally, in Section 4.3 we conclude and discuss future research.

4.1 From Topologic to Communication Graphs

In [64], Moss and Parikh introduce a bimodal logic intended to formalize reasoning about points and sets. This new logic called *Topologic* can also be understood as an epistemic logic with an effort modality. Formally, the two modalities are: K and $\Diamond$. The intended interpretation of $K\phi$ is that ϕ is known; and the intended interpretation of $\Diamond\phi$ is that after some amount of effort ϕ becomes true. For example, the formula

$$\phi \rightarrow \Diamond K\phi$$

means that if ϕ is true, then after some "work", $K\phi$ becomes true, i.e., ϕ is known. In other words, the formula says that if ϕ is true, then ϕ can be known with some effort. What exactly is meant by "effort" depends on the application. For example, we may think of effort as meaning taking a measurement, performing a calculation or observing a computation. In this chapter we will think of effort as meaning consulting some agent's database of known formulas.

There is a temptation to think that the effort modality can be understood as (only) a temporal operator, reading $\Diamond\phi$ as "ϕ is true some time in the future". While there is a

connection between the logics of knowledge and time, and logics of knowledge and effort (see [54, 55] and references therein for more on this topic), following [64] we will assume that such effort leaves the base facts about the world unchanged. In particular, in any topologic model, if ϕ does not contain any modalities, then $\phi \leftrightarrow \Box\phi$ is valid. Thus, effort will not change the base facts about the world – it can only change knowledge of these facts.

The family of logics introduced in [64] and later studied by Dabrowski, Moss and Parikh, Georgatos, Heinemann, and Weiss ([29, 41, 42, 43, 54, 108]) has a semantics in which the acquisition of knowledge is explicitly represented. Familiar mathematical structures such as subset spaces, topologies, intersection spaces and complete lattices of subsets corresponding to natural notions of knowledge acquisition are attached to standard Kripke structures.

Given a set W, a *subset space* is a pair $\langle W, \mathcal{O}\rangle$, where $\mathcal{O}$ is a collection of subsets of W. A point $x \in W$ represents a complete description of the world in which all ground facts are settled, whereas a set $U \in \mathcal{O}$ represents an *observation.* The pair (x, U), called a *neighborhood situation*, can be thought of as an actual situation together with an observation made about the situation. Formulas are interpreted at neighborhood situations. Thus the knowledge modality K represents movement within (consistent with) the current observation, while the effort modality $\Diamond$ represents a refining of the current observation.

Formally a subset model, or a Topologic model, is a triple $\langle W, \mathcal{O}, V \rangle$ where $\langle W, \mathcal{O} \rangle$ is a subset space and V is a valuation function (i.e., a function from atomic propositional variables to subsets of W). The definition of truth of the boolean connectives is as usual. We give the definition of truth of the atomic propositions and the modal operators:

1. $x, U \models p$ iff $x \in V(p)$ where p is an atomic proposition

2. $x, U \models K\phi$ iff for each $y \in U$, $y, U \models \phi$

3. $x, U \models \Diamond\phi$ iff there exist a $V \in \mathcal{O}$ such that $x \in V \subseteq U$ and $x, V \models \phi$.

The following axioms and rules constitute the core axiom system for any Topologic. This axiom system was shown by Moss and Parikh [64] to be sound and complete for all subset spaces.

1. All propositional tautologies and Modus Ponens

2. $(p \rightarrow \Box p) \wedge (\neg p \rightarrow \Box \neg p)$, for $p \in \mathsf{At}$.

3. $\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$

4. $\Box\phi \rightarrow \phi$

5. $\Box\phi \rightarrow \Box\Box\phi$

6. $K_i(\phi \rightarrow \psi) \rightarrow (K_i\phi \rightarrow K_i\psi)$

7. $K_i\phi \rightarrow \phi$

8. $K_i\phi \rightarrow K_iK_i\phi$

9. $\neg K_i\phi \rightarrow K_i\neg K_i\phi$

10. (Cross axiom) $K_i\Box\phi \rightarrow \Box K_i\phi$

11. From ϕ infer $\Box\phi$

12. From ϕ infer $K\phi$

In [41] and [42], Georgatos provides a sound and complete axiomatization for subset spaces that are topological spaces and complete lattices. Dabrowski, Moss, and Parikh prove the same result using an embedding into **S4** [29]. In [43], Georgatos provides a sound and complete axiomatization for treelike spaces, and Weiss [108] has provided a sound and complete axiomatization for intersection-spaces. Interestingly, it is shown in [108] that an infinite number of axiom schemes are necessary for any complete axiomatization of intersection spaces. More recently, Heinemann [54, 55] has looked at subset spaces and logics of knowledge and time, and the connection between hybrid logic and subset spaces [56].

In this paper, we present a multi-agent topologic in which the effort modality $\Diamond$ is intended to mean communication among agents. In order for any communication to take place, we must assume that the agents understand a common language. Thus we assume a set At of propositional variables, understood by all the agents, but with only specific agents knowing their actual values at the start. Letters p, q, etc, will denote

elements of At. The agents will have some information – knowledge of the truth values of some elements of At, but refine that information by acquiring information possessed by other agents, possibly via yet other agents. This implies that if agents are restricted in whom they can communicate with, then this fact will restrict the knowledge they can acquire.

Consider the current situation with Bush and Porter Goss, the director of the CIA. If Bush wants some information from a particular CIA operative, say Bob, he must get this information through Goss. Suppose that ϕ is a formula representing the exact whereabouts of Bin Laden, and that Bob, the CIA operative in charge of maintaining this information knows ϕ. In particular, $K_{\text{Bob}}\phi$, but suppose that at the moment, Bush does not know the exact whereabouts of Bin Laden ($\neg K_{\text{Bush}}\phi$). Presumably Bush *can* find out the exact whereabouts of Bin Laden ($\Diamond K_{\text{Bush}}\phi$) by going through Goss, but of course, *we* cannot find out such information ($\neg\Diamond K_{\text{e}}\phi \wedge \neg\Diamond K_{\text{r}}\phi$) since we do not have the appropriate security clearance. Clearly, then, as a *pre-requisite* for Bush learning ϕ, Goss will also have come to know ϕ. We can represent this situation by the following formula:

$$\neg K_{\text{Bush}}\phi \wedge \Box(K_{\text{Bush}}\phi \rightarrow K_{\text{Goss}}\phi)$$

where $\Box$ is the dual of diamond.

Let $\mathcal{A}$ be a set of agents. A **communication graph** is a directed graph $\mathcal{G}_{\mathcal{A}} = (\mathcal{A}, E)$ where $E \subseteq \mathcal{A} \times \mathcal{A}$. Intuitively $(i, j) \in E$ means that i can directly receive information

from agent j, but *without* j knowing this fact. Thus an edge between i and j in the communication graph represents a one-sided relationship between i and j. Agent i has access to any piece of information that agent j knows. We have introduced this 'one sidedness' restriction in order to simplify our semantics, but also because such situations of one sided learning occur naturally. A common situation that is helpful to keep in mind is accessing a website. We can think of agent j as creating a website with limited access in which everything he *currently* knows is available, and then if there is an edge between i and j then agent i can access this website without j being aware that the site is being accessed. Another important application is spying where one person accesses another's information without the latter being aware that information is being leaked. Naturally j may have been able to access some other agent k's website and had updated some of her own information. Therefore, it is important to stress that when i accesses j's website, he is accessing j's current information which may include what k knew initially.

The assumption that i can access all of j's information is a significant idealization from these common situations, but becomes more realistic if we think of this information as being confined to facts expressible as truth functional combinations of some small set of basic propositions. Thus our idealization rests on two assumptions:

1. All the agents share a common language, and

2. The agents make available all possible pieces of information which they know *and which are expressible in this common language.*

4.2 The Logic of Communication Graphs

In this section we will describe the logic of communication graphs, $\mathcal{K}(\mathcal{G})$. The language will be a multi-agent modal language with a communication modality. The formula $K_i\phi$ will be interpreted as "according to i's current information, i knows ϕ", and $\Diamond\phi$ will be interpreted as "after some communications (which respect the communication graph), ϕ becomes true". Thus for example, the multi-agent version of the formula $\phi \rightarrow \Diamond K\phi$, expressing that if ϕ is true then with some effort ϕ can be known, is

$$K_j\phi \rightarrow \Diamond K_i\phi$$

This formula expresses that if agent j (currently) knows ϕ, then after some communication, agent i can come to know ϕ. Let At be a finite set of propositional variables. Well-formed formulas of $\mathcal{K}(\mathcal{G})$ are generated according to the following grammar

$$\phi \ := p \mid \neg\psi \mid \phi \wedge \psi \mid K_i\phi \mid \Diamond\phi$$

where $p \in$ At. We abbreviate $\neg K_i\neg\phi$ and $\neg\Diamond\neg\phi$ by $L_i\phi$ and $\Box\phi$ respectively, and use the standard abbreviations for the propositional connectives ($\vee$, $\rightarrow$, and $\perp$). Let $\mathcal{L}_{\mathcal{K}(\mathcal{G})}$ denote the set of well-formed formulas of $\mathcal{K}(\mathcal{G})$. We also define $\mathcal{L}_0$(At), (or simply $\mathcal{L}_0$ if At is fixed or understood), to be the set of ground formulas, i.e., the set of formulas

constructed from At using $\neg$ and $\wedge$ only.

4.2.1 Semantics

The semantics presented here combines ideas both from the subset models of described in the previous section and the history based models of Parikh and Ramanajum (see [82, 83] and Chapter 2). Suppose that $\mathcal{G} = (\mathcal{A}, E_{\mathcal{G}})$ is a fixed communication graph. Given that the agents are initially given some private information and assumed to communicate according to the communication graph $\mathcal{G}$, the semantics in this section is intended to formalize what agents know and may come to know after some communication.

Initially, each agent i knows or is informed (say by nature) of the truth values of a certain subset At_i of propositional variables, and the At_i *as well as this fact are common knowledge.* Thus the other agents know that i knows the truth values of elements of At_i, but, typically, not what these values actually are. We do not need to assume that the At_i are disjoint, nor that the At_i together add up to all of At, although such sub-cases will be of interest. Thus if At_i and At_j intersect then agents i, j will share information at the very beginning. Let W be the set of boolean valuations on At. An element $v \in W$ is called a *state.* We use 1 for the truth value *true.* Initially each agent i is given a boolean valuation $v_i : \mathsf{At}_i \to \{0, 1\}$. This initial distribution of information among the agents can be represented by a vector $\vec{v} = (v_1, \ldots, v_n)$. Of course, since we are modeling knowledge and not belief, these initial boolean valuations must be compatible. I.e., for each i, j, v_i

and v_j agree on $\mathsf{At}_i \cap \mathsf{At}_j$. Call any vector of partial boolean valuations $\vec{v} = (v_1, \ldots, v_n)$ **consistent** if for each $p \in dom(v_i) \cap dom(v_j)$, $v_i(p) = v_j(p)$ for all $i, j = 1, \ldots, n$. We shall assume that only such consistent vectors arise as initial information. All this information is common knowledge and only the precise values of the v_i are private.

Definition 16 *Let* At *be a finite set of propositional variables and* $\mathcal{A} = \{1, \ldots, n\}$ *a finite set of agents. Given the distribution of sublanguages* $\vec{\mathsf{At}} = (\mathsf{At}_1, \ldots, \mathsf{At}_n)$, *an* **initial information vector for** $\vec{\mathsf{At}}$ *is any consistent vector* $\vec{v} = (v_1, \ldots, v_n)$ *of partial boolean valuations such that for each* $i \in \mathcal{A}$, $dom(v_i) = \mathsf{At}_i$.

We assumed that all initial vectors are consistent, although if we were dealing with beliefs rather than knowledge, then very interesting questions about *in*-consistent initial vectors could arise.

We assume that the only communications that take place are about the physical world. But we do allow agents to learn objective facts which are not atomic, but may be complex, like $p \vee q$ where $p, q \in \mathsf{At}$. Now note that if agent i learned some *literal* from agent j, then there is a simple way to update i's valuation v_i with this new information by just adding the truth value of another propositional symbol. However, if i learns a more general ground formula from agent j, then the situation will be more complex. For instance if the agent knows p and learns $q \vee r$ then the agent now has three valuations on the set $\{p, q, r\}$ which cannot be described in terms of a partial valuation on a subset of At.

Fix a communication graph $\mathcal{G}$ and suppose that agent i learns some ground fact ϕ from agent j. Of course, there must be an edge from agent i to agent j in $\mathcal{G}$. This situation will be represented by the tuple (i, j, ϕ) and will be called a **communication event**. Let $\Sigma_{\mathcal{G}}$ be the set of all possible events. Formally,

Definition 17 *Let $\mathcal{G} = (\mathcal{A}, E_{\mathcal{G}})$ be a communication graph. A tuple (i, j, ϕ), where $\phi \in \mathcal{L}_0(\mathsf{At})$ and $(i, j) \in E_{\mathcal{G}}$ is called a* **communication event**. *Then $\Sigma_{\mathcal{G}} = \{(i, j, \phi) \mid \phi \in \mathcal{L}_0, (i, j) \in E_{\mathcal{G}}\}$ is the set of all possible communication events (given the communication graph $\mathcal{G}$).*

Given the set of events $\Sigma_{\mathcal{G}}$, a **history** is a finite sequence of events. I.e., $H \in \Sigma_{\mathcal{G}}^*$. The empty history will be denoted ϵ. We remind the reader of the relevant notions from Chapter 2. Given two histories H, H', say $H \preceq H'$ iff $H' = HH''$ for some history H'', i.e., H is an initial segment of H'. Obviously, $\preceq$ is a partial order. If H is a history, and (i, j, ϕ) is a communication event, then H followed by (i, j, ϕ) will be written $H; (i, j, \phi)$. Given a history H, let $\lambda_i(H)$ be i's local history corresponding to H. I.e., $\lambda_i(H)$ is a sequence of events that i can "see". Formally, λ_i maps each event of the form (i, j, ϕ) to itself, and maps other events (m, j, ψ) with $m \neq i$ to the null character while preserving the order among events. Thus we are assuming that the agent's have perfect recall but do not all have access to the global clock.

Fix a finite set of agents $\mathcal{A} = \{1, \ldots, n\}$ and a finite set of propositional variable At along with subsets $(\mathsf{At}_1, ..., \mathsf{At}_n)$. A **communication graph frame** is a pair $\langle \mathcal{G}, \vec{\mathsf{At}} \rangle$

where $\mathcal{G}$ is a communication graph, and $\vec{\mathsf{At}} = (\mathsf{At}_1, ..., \mathsf{At}_n)$ is an assignment of sublanguages to the agents. A **communication graph model** based on a frame $\langle \mathcal{G}, \vec{\mathsf{At}} \rangle$ is a triple $\langle \mathcal{G}, \vec{\mathsf{At}}, \vec{v} \rangle$, where $\vec{v}$ is a consistent vector of partial boolean valuations for $\vec{\mathsf{At}}$.

Now we address two issues. One is that not all histories are legal. For an event (i, j, ϕ) to take place after a history H, it must be the case that after H, j knows ϕ. Clearly i cannot learn from j something which j did not know. Whether a history is justified depends not only on the initial valuation, but also on the set of communications that have taken place prior to each communication in the history.

The second issue is that the information which an agent learns by "reading" a formula ϕ may be *more* than just the fact that ϕ is true. For suppose that i learns $p \vee q$ from j, but j is not connected, directly or indirectly, to anyone who might know the initial truth value of q. In this case i has learned *more* than $p \vee q$, i has learned p as well. For the only way that j could have known $p \vee q$ is if j knew p in which case p must be true. Our definition of the semantics below will address both these issues.

Formulas will be interpreted at pairs (w, H) where w is a state (boolean valuation) and H is a finite sequence of communication events.

We first introduce the notion of i-equivalence among histories. Intuitively, two histories are i-equivalent if those communications which i takes active part in, are the same.

Definition 18 *Let w be a state and H a finite history. Define the relation $\sim_i$ as follows:*

$(w, H) \sim_i (v, H')$ *iff* $w_{|At_i} = v_{|At_i}$ *and* $\lambda_i(H) = \lambda_i(H')$.

Before proceeding further, we summarize the uncertainty faced by each of the agents:

1. Agents may be uncertain about the actual state of the world.

2. Agents may be uncertain about which communications have taken place.

Example 4.2.1 The Valerie Plame Affair: *In an earlier version of this paper we stated that if a formula ϕ was stable, agent j knew it, and agent i was connected either directly or indirectly to agent j, then agent i could also come to know ϕ. Here a formula ϕ is said to be stable if for all legal (w, H), $(w, H) \models_{\mathcal{M}} (\phi \rightarrow \Box\phi)$.*

However, we were mistaken and an abstract example as well as the Valerie Plame/Judith Miller affair shows why. Suppose that agent i is connected directly to agent j who is connected directly to agents k, m, both of whom are connected to r who knows the value of p. Now m reads p, which is true, from r's website, and j reads p from m's website and thus knows not only that p but also $K_m(p)$. Now the formula $K_m(p)$ is stable, it will never again become false. But i cannot know this although i can know p. For just by reading j's web page, *i cannot rule out the possibility that j learned about p from k.*

The way in which this applies to the Plame-Miller affair is that the fact that Plame was a CIA covert operative was revealed by columnist Robert Novak in July 2003, possibly endangering her life, and this information seems to have come from Miller who is under a federal sentence for refusing to reveal who leaked the name of Valerie Plame to Novak.

The point here is that while we know what *Miller and Novak knew about Plame, we do not know* how *they knew it.*

To deal with the notion of legal or justified history we introduce a propositional symbol L which is satisfied only by legal pairs (w, H). (We may also write $L(w, H)$ to indicate that the pair (w, H) is legal.) Since L can only be defined in terms of knowledge, and knowledge in turn requires quantification over legal histories we shall need mutual recursion.

Given a communication graph and the corresponding model $\mathcal{M} = \langle \mathcal{G}, \vec{\mathsf{At}}, \vec{v} \rangle$, and pair (w, H), we define the legality of (w, H) and the truth $\models_{\mathcal{M}}$ of a formula as follows:

- $w, \epsilon \models_{\mathcal{M}} L$
- $w, H; (i, j, \phi) \models_{\mathcal{M}} L$ iff $w, H \models_{\mathcal{M}} L$ and $w, H \models_{\mathcal{M}} K_j \phi$
- $w, H \models_{\mathcal{M}} p$ iff $w(p) = 1$, where $p \in \mathsf{At}$
- $w, H \models_{\mathcal{M}} \neg\phi$ iff $w, H \not\models_{\mathcal{M}} \phi$
- $w, H \models_{\mathcal{M}} \phi \wedge \psi$ iff $w, H \models_{\mathcal{M}} \phi$ and $w, H \models_{\mathcal{M}} \psi$
- $w, H \models_{\mathcal{M}} \Diamond\phi$ iff $\exists H'$, $H \preceq H'$, $L(w, H')$, and $w, H' \models_{\mathcal{M}} \phi$
- $w, H \models_{\mathcal{M}} K_i \phi$ iff $\forall (v, H')$ if $(w, H) \sim_i (v, H')$, and $L(v, H')$, then $v, H' \models_{\mathcal{M}} \phi$

Unless otherwise stated, we will only consider legal pairs (w, H), i.e., pairs (w, H) such that $w, H \models L$. We say ϕ is **valid in** $\mathcal{M}$, $\models_{\mathcal{M}} \phi$ if for all (w, H), $w, H \models_{\mathcal{M}} \phi$. ϕ is **valid in the communication graph frame** $\mathcal{F}$ if ϕ is valid in all models based on $\mathcal{F}$.

4.2.2 Surface Knowledge

Except for each agent's initial information, one may suspect that all information acquired by the agent i is just the sum of the ϕ which i learned from communications (i, j, ϕ). But we saw that this is not true. Given the assumption that both $\vec{\mathrm{At}}$ and the structure of the communication graph are common knowledge, agents can come to know facts that are not explicitly contained in the communications. We might still be interested in this 'surface' knowledge which the agents acquire.

Define the sets $X_i(w, H)$ as follows:

1. $X_i(w, \epsilon) = \{v \mid v_{|\mathrm{At}_i} = w_{|\mathrm{At}_i}\}$
2. $X_i(w, H; (i, j, \phi)) = X_i(w, H) \cap \hat{\phi}$
3. $i \neq m$ then $X_i(w, H; (m, j, \phi)) = X_i(w, H)$

Intuitively, if $X_i(w, H) \subseteq \widehat{\phi}$, then ϕ is implied (for i) by the sequence of communications. We first show a preliminary lemma which is needed to show that at (w, H), agents know at least the formulas implied by $X_i(w, H)$.

Lemma 4.2.1 *If* $(w, H) \sim_i (v, H')$, *then* $X_i(w, H) = X_i(v, H')$.

Proof The proof is by induction on $\lambda_i(H) = \lambda_i(H')$. If $\lambda_i(H)$ was empty then H itself might as well be ϵ, and then we use the fact that $X_i(w, \epsilon) = \{u \mid u_{|\mathsf{At}_i} = w_{|\mathsf{At}_i}\}$ is the same as $X_i(v, \epsilon) = \{u \mid u_{|\mathsf{At}_i} = v_{|\mathsf{At}_i}\}$ since $w_{|\mathsf{At}_i} = v_{|\mathsf{At}_i}$. Otherwise we use the fact that since $\lambda_i(H) = \lambda_i(H')$, the initial set $X_i(w, \epsilon) = X_i(v, \epsilon)$ went through exactly the same intersections with various $\hat{\phi}$ when the ground facts ϕ were learned by i. Indeed $X_i(w, H)$ depends *only* on the *set* of ϕ which i learned in H and not on their order. In particular, If (i, j, ϕ) already occurs in H, then $X_i(w, H; (i, j, \phi)) = X_i(w, H)$. □

Lemma 4.2.2 *Let $\mathcal{M} = \langle \mathcal{G}, \vec{\mathsf{At}}, \vec{v} \rangle$ be any communication graph model and ϕ a ground formula. If $X_i(w, H) \subseteq \widehat{\phi}$, then $(w, H) \models_{\mathcal{M}} K_i(\phi)$.*

Proof Let $\mathcal{M} = \langle \mathcal{G}, \vec{\mathsf{At}}, \vec{v} \rangle$ be a communication graph model. Suppose that ϕ is a ground formula with $X_i(w, H) \subseteq \widehat{\phi}$. Let $(v, H') \sim_i (w, H)$. We must show that $v, H' \models_{\mathcal{M}} \phi$. Since ϕ is a ground formula, this is equivalent to showing that $v(\phi) = 1$. Since $(w, H) \sim_i (v, H')$ by Lemma 4.2.1 $X_i(v, H') = X_i(w, H) \subseteq \widehat{\phi}$. Thus we need only the following claim.

Claim: If $X_i(v, H') \subseteq \widehat{\phi}$, then $v(\phi) = 1$.

Proof of claim: The proof is by induction on H'. If $H' = \epsilon$, then since $X_i(v, H') = \{y \mid y_{|\mathsf{At}_i} = v_{|\mathsf{At}_i}\}$ and, of course, $v_{\mathsf{At}_i} = v_{|\mathsf{At}_i}$, we have $v \in X_i(v, H') \subseteq \widehat{\phi}$. Hence $v(\phi) = 1$.

Suppose that $m \neq i$ and $H' = H_1; (m, j, \psi)$. Then by construction $X_i(v, H') = X_i(v, H_1)$, and so, since $X_i(v, H_1) = X_i(v, H') \subseteq \widehat{\phi}$, by the induction hypothesis we have $v(\phi) = 1$.

Finally suppose that $H' = H_1(i, j, \psi)$. Then $X_i(v, H') = X_i(v, H_1) \cap \widehat{\psi}$. Since we only consider justified state-history pairs, $X_j(v, H_1) \subseteq \widehat{\psi}$. Hence, by the induction hypothesis $v(\psi) = 1$. Let θ be any formula such that $X_i(v, H_1) = \widehat{\theta}$ (such a formula must exist since At is finite and so every set of states can be defined by a formula). By the induction hypothesis since $X_i(v, H_1) = \widehat{\theta}$, $v(\theta) = 1$. Hence $\widehat{\theta} \cap \widehat{\psi} = X_i(v, H_1; (i, j, \psi)) \subseteq \widehat{\phi}$. Since $v(\theta) = v(\psi) = 1$, $v(\phi) = 1$. This completes the proof of the claim and of the lemma. □

But as we saw, the converse is not true. That is, there are ground formulas that the agents may come to know that are not explicitly contained in their communications. Essentially, these are facts that the agents can derive given their knowledge of the structure of the communication graph and the initial distribution of facts. The sets $X_i(w, H)$ represent the knowledge which agents i would acquire after communication *if* they did not know the structure of the graph.

4.2.3 Axioms and Decidability

The following axioms and rules are known to be sound and complete with respect to the set of all subset spaces ([64]). Thus they represent the core set of axioms and rules for any topologic.

1. All propositional tautologies
2. $(p \to \Box p) \wedge (\neg p \to \Box \neg p)$, for $p \in \mathsf{At}$.
3. $\Box(\phi \to \psi) \to (\Box\phi \to \Box\psi)$
4. $\Box\phi \to \phi$
5. $\Box\phi \to \Box\Box\phi$
6. $K_i(\phi \to \psi) \to (K_i\phi \to K_i\psi)$
7. $K_i\phi \to \phi$
8. $K_i\phi \to K_iK_i\phi$
9. $\neg K_i\phi \to K_i\neg K_i\phi$
10. (Cross axiom) $K_i\Box\phi \to \Box K_i\phi$

We include the following rules: modus ponens, K_i and $\Box$ necessitation. We write $\vdash \phi$ if ϕ can be derived from any of the above schemes and rules. The soundness of axioms 1-9 and the rules are easy to verify also for our framework.

We now show that the cross axiom $K_i\Box\phi \to \Box K_i\phi$ is sound. It is easier to consider it in its contrapositive form: $\Diamond L_i\phi \to L_i\Diamond\phi$. This is interpreted as follows: if there is a sequence of updates that lead agent i to consider ϕ possible, then i already thinks it possible that there is a sequence of updates after which ϕ becomes true.

Proposition 4.2.3 *$\Diamond L_i\phi \rightarrow L_i\Diamond\phi$ is valid in all communication graph models.*

Proof Let $\mathcal{M} = \langle \mathcal{G}, \vec{\mathsf{At}}, \vec{v} \rangle$ be a communication graph model and (w, H) any justified state-history pair. Suppose that $w, H \models \Diamond L_i\phi$. Then there exists H' with $H \preceq H'$ such that $w, H' \models L_i\phi$. Hence there is a pair (v, H'') such that $(v, H') \sim_i (w, H'')$ and $v, H'' \models_{\mathcal{M}} \phi$. Let H''' be any sequence such that $\lambda_i(H) = \lambda_i(H''')$ and $H''' \preceq H''$. Such a history must exist since $H \preceq H'$ and $H' \sim_i H''$. Since $H \preceq H'$, $\lambda_i(H) \preceq \lambda_i(H') = \lambda_i(H'')$. Therefore, we need only let H''' be any initial segment of H'' containing $\lambda_i(H)$. By definition of L, all initial sequences of a legal history are legal. Therefore, since $v, H'' \models_{\mathcal{M}} \phi$, $v, H''' \models \Diamond\phi$; and since $H \sim_i H'''$, $w, H \models_{\mathcal{M}} L_i\Diamond\phi$. □

We leave the problem of finding a complete axiomatization for a future paper, and move to decidability. We show that the satisfiability problem is decidable by showing that a satisfiable formula has a model of bounded size. The main idea is to show that for any history H in which an event of the form (i, j, ϕ) occurs twice is "equivalent" to another history in which that event only occurs once. Here "equivalent" means satisfies the same formulas. We first need a definition. Given any history H, let $f(H)$ be the sequence of events of H generated by the order: e comes before e' iff the first occurrence of e in H occurred before the first occurrence of e' in H. Thus $f(H)$ is the compressed history obtained from H by deleting the second and subsequent occurrences of any event. Thus, for instance, if $H = e_2e_1e_2e_1e_3$ then $f(H) = e_2e_1e_3$.

Definition 19 *Let $w \in W$ be any state and suppose that H and H' are justified histories (for w). We say that H and H' are C-equivalent, written $C(H, H')$, iff $f(H) = f(H')$.*

Intuitively, for two histories H and H', $C(H, H')$ holds if their compressed versions are the same.

Lemma 4.2.4 *Fix a state w and suppose that H and H' are justified histories. Then*

1. *If $C(H, H')$ and $L(w, HH_1)$, then $L(w, H'H_1)$ and $C(HH_1, H'H_1)$.*
2. *If $C(H, H')$ and $H \sim_i H_1$ for some i, then there is a legal history H_1' such that $C(H_1, H_1')$ and $H' \sim_i H_1'$.*

Proof Let w be a state and H and H' two justified histories such that $C(H, H')$. To prove part 1, Let H_1 be any history such that HH_1 is legal. Now the legality of an event (i, j, ϕ) in H_1 as part of HH_1 depended on the fact that j knew ϕ. Now every (j, m, ψ) which occurred in H also occurred in H' and if it occurred in H_1 as part of HH_1 it would also occur in H_1 as part of $H'H_1$. Thus the same justifications for H_1 events are available in both cases and $H'H_1$ must also be legal. Clearly, $f(HH_1) = f(H'H_1)$. Therefore $C(HH_1, H'H_1)$.

For part 2, suppose that $H \sim_i H_1$ for some agent i and legal history H_1. Since $H \sim_i H_1$, $\lambda_i(f(H)) = \lambda_i(f(H_1))$. Also, since $f(H) = f(H')$, $\lambda_i(f(H)) = \lambda_i(f(H'))$. Therefore, $\lambda_i(f(H')) = \lambda_i(f(H_1))$.

That is, the sequence of first occurrence of i events in H' is the same as the sequence of first occurrence of i events in H_1. Thus, by adding extra i events to or removing excess i events from H_1, a history H_1' can be constructed such that $H' \sim_i H_1'$. Clearly by construction $f(H_1) = f(H_1')$. □

The follow Corollaries are easy consequences of the above Lemma and so the proofs will be omitted.

Corollary 4.2.5 *Let the relation D between state history pairs be defined by*

$D((w,H),(w,H'))$ *iff* $C(H,H')$. *Then D is a bisimulation.*

Corollary 4.2.6 *If H contains (i,j,ϕ) and $L(w,H)$ holds, then also $L(H;(i,j,\phi))$, and for all ψ, $(w,H) \models \psi$ iff $(w,H;(i,j,\phi)) \models \psi$*

Corollary 4.2.7 *If a formula ϕ is satisfiable in some graph model $(\mathcal{G},\vec{\mathsf{At}})$ then it is satisfiable in a history in which no communication (i,j,ϕ) occurs twice.*

This last result immediately gives us a decision procedure as we can limit the length of the history which might satisfy some given formula ϕ. Now there are only a finite number of ground formulas ϕ, thus only a finite number of learnings (i,j,ϕ), and hence only a finite number of histories we need to look at. Alas, this number is quite large and we hope to find a better decision procedure. Note that if we limited the agents

to read *only* atomic formulas, a very natural restriction, then the number of possible communications would be smaller and the decision procedure would be faster, and indeed would be in non-deterministic exponential time. The logic *would* change as the formulas $K_i(p \vee q) \rightarrow K_i(p) \vee K_i(q)$ would be valid with such a restriction, but are not valid if non-atomic formulas can be read from another agent's website.

We now define a maximal history (relative to some w) as a history in which all possible (finitely many) communication events have taken place at least once. If H is a maximal history, then we will have, for all H', $C(H, HH')$ and hence for all H', all w, ϕ, $w, H \models \phi$ iff $w, HH' \models \phi$. In other words, a maximal w, H satisfies, for all ϕ, $\phi \leftrightarrow \Box\phi$.

Lemma 4.2.8 *The axiom $\Box\Diamond\phi \rightarrow \Diamond\Box\phi$ is valid in Logic of Communication Graphs.*

Proof Fix w compatible with some history H which satisfies $\Box\Diamond\phi$. Let H' be a maximal history extending H, then w, H' satisfies $\Diamond\phi$ and hence ϕ and hence $\Box\phi$. Since H' extends H, w, H satisfies $\Diamond\Box\phi$. □

We strongly suspect that if H and H' are maximal histories (relative to w), then w, H and w, H' satisfy the same formulas. In this case, $\Diamond\Box\phi \rightarrow \Box\Diamond\phi$ would be valid. This and other issues related to a complete axiomatization will be left for another paper.

4.2.4 Connection with Communication Graphs

In this section we will investigate the close connection between formulas valid in a model based on the communication graph and the communication graph. We will prove that the valid formulas characterize the communication graph.

Theorem 4.2.9 *Let $\mathcal{G} = (\mathcal{A}, E)$ be a communication graph. Then $(i, j) \in E$ if and only if, for all $l \in \mathcal{A}$ such that $l \neq i$ and $l \neq j$ and all ground formulas ϕ, the scheme*

$$K_j\phi \wedge \neg K_l\phi \rightarrow \Diamond(K_i\phi \wedge \neg K_l\phi)$$

is valid in all communication graph models based on $\mathcal{G}$.

Proof Suppose that $w, H \models_M K_j\phi \wedge \neg K_l\phi$. Then j knows ϕ and hence i can read ϕ directly from j's website. l is none the wiser as $\lambda_l(H) = \lambda_l(H; (i, j, \phi))$. Therefore, $w, H; (i, j, \phi) \models K_i\phi \wedge \neg K_l\phi$. □

4.3 Conclusions and Further Work

In this chapter we have introduced a logic of knowledge and communication. Communication among agents is restricted by a communication graph, and idealized in the sense that the agents are unaware when their knowledge base is being accessed. We have

shown that the communication graph is characterized by the validities of formulas in models based on that communication graph, and that our logic is decidable.

As discussed in the introduction, logics of knowledge acquisition through communication have been the focus of recent study. We can now be more precise about the similarities and differences between our approaches. These logics use **PDL** style operators to represent an epistemic update. For example, if $!\phi$ is intended to mean a public announcement of ϕ, then $\langle !\phi \rangle K_i \phi$ is intended to mean that after ϕ is publicly announced, agent i knows ϕ. From this point of view, the communication modality $\Diamond$ can be understood as existentially quantifying over a sequence of private epistemic updates. However, there are some important differences between the semantics presented in this paper and the semantics found in the dynamic epistemic logic literature. First of all, in our semantics, communication is limited by the communication graph. Secondly, we do not consider general epistemic updates as is common in the literature, but rather study a specific type of epistemic update and its connection with a communication graph. Most important is the fact that the history of communications plays a key role in the definition of knowledge in this paper. The general approach of dynamic epistemic semantics is to define update operations mapping Kripke structures to other Kripke structures intended to represent the effect of an epistemic update on the first Kripke structure. For example, a public announcement of ϕ selects the submodel of a Kripke structure in which ϕ is true at every state. The definition of knowledge after an epistemic update is the usual

definition, i.e., ϕ is known by i at state w if ϕ is true in all states that i considers possible from state w in the updated Kripke structure.

Notice that if we restrict our attention to maximal histories, then the following property will be satisfied: For any two agent i and j if there is a path in the communication graph from i to j then any ground fact that j knows, i will also know. In this case, we can say that j *dominates* i. This notion of domination was studied by Fitting in [35, 36]. Fitting develops a semantics in which there are multiple agents and a domination relation on the set of agents (which is assumed to be a partial order). He studies the question of how to assign truth values to formulas in a common modal language and on which sentences will the agents agree. Two semantics are offered. One is a combination of Kripke intuitionistic models and Kripke modal models and the second is a many-valued Kripke modal model. The two semantics are shown to be equivalent and a sound and complete axiomatization is offered.

Moving on to future work. Standard questions such as finding an elegant complete axiomatization will be studied. Another interesting extension would be to allow different types of updates, such as lying, conscious updates (where j is aware that his website is being read), updating to subgroups (creating common knowledge) and so on.

Another natural extension is to consider situations in which agents have a preference over which information they will read from another agent's website. Thus for example, if one hears that an English Ph.D. student and his adviser recently had a meeting,

then one is justified in assuming that they probably did not discuss the existence of non-recursive sets, even though the adviser may conceivably know this fact. Given that this preference over the formulas under discussion among different groups of agents is common knowledge, each agent can regard some (legal) histories as being more or less likely than other (legal) histories. From this ordering over histories, we can define a defeasible knowledge operator for each agent. The operator is defeasible in the sense that agents may be wrong, i.e., it *is* after all possible that the English student and his adviser actually spent the meeting discussing the fact that there must be a non-recursive set.

Finally we remark that our framework and the logic can be seen as a demonstration of the need for cryptographic protocols. Two issues are important here. The first is that an agent may only want part of its knowledge base to be accessible by the public. This may be modeled in our framework by restricting for each agent j the set of formulas that the agent makes available, and so when i is directly connected to j, i can only update by facts in the accessible domain. The second issue is that we may not know the exact structure of the communication graph. For example, if Ann accesses some information from Bob's website, but unknown to Ann, Charles is listening in, then the communication graph has an edge between Charles and Bob, whose presence is not known to Ann or to Bob. Then clearly as a condition for Ann learning some information from Bob, Charles must be able to be informed of that same piece of information. Thus

cryptographic protocols are essentially intended to ensure that there are no "undesired edges" between agents in the communication graph. Moreover, in that version of our model where the entire graph is not common knowledge, inferring the existence of edges *from* knowledge is yet another, potentially important extension.

Chapter 5

Strategic Voting

Whether made explicit or implicit, knowledge theoretic properties such as common knowledge of rationality are important in understanding and modeling game-theoretic, or strategic, situations. There is a large literature devoted to exploring these and other issues related to the epistemic foundations of game theory. The recent paper [19] and the survey [11] have excellent discussions and pointers to the relevant literature. Much of the literature focuses on what the agents need to know about the other agents' strategies, rationality or knowledge in order to guarantee that a particular solution concept, such as the Nash equilibrium, is realized. In other words, what are the knowledge-theoretic properties that make a particular solution concept "effective". In [23], together with Samir Chopra and Rohit Parikh, we develop a framework that looks at similar issues relevant to the field of voting theory. This chapter is based on [23]. Our analysis suggests

that an agent must possess information about the other agents' preferences in order for the agent to decide to vote strategically. In a sense, our claim is that the agents need a certain amount of information in order for the Gibbard-Satterthwaite theorem to be "effective".

5.1 Introduction

A comprehensive theory of multi-agent interactions must pay attention to results in social choice theory such as the Arrow and Gibbard-Satterthwaite theorems [2, 45, 95]. These impossibility results constrain the existence of rational collective decision making procedures. In this chapter we turn our attention to another aspect of social aggregation scenarios: the role played by the states of knowledge of the agents. The study of strategic interactions in game theory reflects the importance of states of knowledge of the players. In this chapter, we bring these three issues—states of knowledge, strategic interaction and social aggregation operations—together.

The Gibbard-Satterthwaite theorem is best explained as follows. Let S be a social choice function whose domain is an n-tuple of preferences $P_1 \ldots P_n$, where $\{1, ..., n\}$ are the voters, $\mathcal{O}$ is the set of choices or candidates and each P_i is a linear order over $\mathcal{O}$. S takes $P_1 \ldots P_n$ as input and produces some element of $\mathcal{O}$ - the winner. Then the theorem says that there must be situations where it 'profits' a voter to vote *strategically*. Specifically, if P denotes the actual preference ordering of voter i, Y denotes the profile

consisting of the preference orderings of all the other voters then the theorem says that there must exist P, Y, P' such that $S(P', Y) >_P S(P, Y)$. Here $>_P$ indicates: better according to P. Thus in the situation where the voter's actual ordering is P and all the orderings of the other voters (together) are Y then voter i is better off saying its ordering is P' rather than what it actually is, namely P. In particular, if the vote consists of voting for the highest element of the preference ordering, it should vote for the different highest element of P' rather than of P.

Of course, the agent might be *forced* to express a different preference. For example, if an agent, whose preferences are $B > C > A$, is only presented C, A as choices, then the agent will pick C. This 'vote' differs from the agent's true preference, but should not be understood as 'strategizing' in the true sense.

A real-life example of strategizing was noticed in the 2000 US elections when some supporters of Ralph Nader voted for their second preference, Gore,[1] in a vain attempt to prevent the election of George W. Bush. Similar examples of strategizing have occurred in other electoral systems over the years ([17] may be consulted for further details on the application of game-theoretic concepts to voting scenarios). The Gibbard-Satterthwaite theorem points out that situations like the one pointed out above *must* arise.

What interests us are the *knowledge-theoretic properties* of the situation described

[1]Surveys show that had Nader not run, 46% of those who voted for him would have voted for Gore, 23% for Bush and 31% would have abstained. Hereafter, when we refer to Nader voters we shall mean those Nader voters who did or would have voted for Gore.

above. We note that unless the voter with preference P *knows* that it should vote strategically, and how, i.e., knows that the other voters' preference is Y and that it should vote according to $P' \neq P$, the theorem is not 'effective'. That is, the theorem only applies in those situations where a certain level of knowledge exists amongst voters. Voters completely or partially ignorant about other voters' preferences would have little incentive to change their actual preference at election time. In the 2000 US elections, many Nader voters changed their votes *because* opinion polls had made it clear that Nader stood no chance of winning, and that Gore could lose as a result of their votes going to Nader.

The goal of this chapter is to propose a formal model in which the effect of poll information on an agent's choice of a vote can be studied. The need for such a model was suggested by Brams and Fishburn in Chapter 7 of [16]. In particular, we are interested in formally showing how voters use poll information during an election. There is a large literature which studies strategic voting in the presence of poll information. As much of the literature is geared towards a political science audience, we only discuss the papers which are related to the goals of this chapter. For a discussion of formal voting theory see [17, 18]. The discussion found in Chapter 7 of [16] has much in common with this chapter and so will be discussed in more detail below. For a overview of models of strategic voting in complete information environments, see [91, 68, 69]. Taking a more "computer science" approach, [26, 27, 25] provides a series of results concerning how

"hard"[2] it is to take advantage of poll information. The articles [20] and [32], which compare sequential voting to simultaneous voting, both discuss issues relevant to this chapter. Finally, the reader is referred to [28] for a discussion of a voting procedure, called *declared-strategy voting*, which attempts to curtail the effects of strategic voting on an election.

5.2 A Formal Voting Model

There is a wealth of literature on formal voting theory. This section draws upon discussions in [17, 18]. The reader is urged to consult these for further details.

Let $\mathcal{O} = \{o_1, \ldots, o_m\}$ be a set of candidates, $\mathcal{A} = \{1, \ldots, n\}$ be a set of agents or voters. We assume that each voter has a preference over the elements of $\mathcal{O}$, i.e., a reflexive, transitive and connected relation on $\mathcal{O}$. For simplicity we assume that each voter's preference is strict. A voter i's *strict preference relation* on $\mathcal{O}$ will be denoted by P_i. We assume that each P_i is a complete, reflexive, transitive and anti-symmetric binary relation on $\mathcal{O}$. For two candidate $o, v \in \mathcal{O}$, we will write $o >_{P_i} v$ iff $(o, v) \in P_i$ and say that i strictly prefers o to v. Henceforth, for ease of readability we will use **Pref** to denote preferences over $\mathcal{O}$. A *preference profile* is an element of $(\mathbf{Pref})^n$.

In voting scenarios such as elections, agents are not expected to announce their actual preference relation, but rather to select a vote that 'represents' their preference.

[2]Here "hard" is being used technically: the results are complexity theoretic.

Each voter chooses a vote v, an aggregation function tallies the votes of each candidate and selects a winner (or winners if electing more than one candidate). There are two components to any voting procedure. First, the type of votes that voters can cast. For example, in *plurality voting* voters can only vote for a single candidate so votes v are simply singleton subsets of $\mathcal{O}$, whereas in *approval voting* voters select a set of candidates so votes v are any subset of $\mathcal{O}$. Following [18], given a set of $\mathcal{O}$ of candidates, let $\mathcal{B}(\mathcal{O})$ be the set of feasible votes, or *ballots*. The second component of any voting procedure is the way in which the votes are tallied to produce a winner (or winners if electing more than one candidate). We assume that the voting aggregation function will select exactly one winner, so ties are always broken[3]. Note that elements of the set $\mathcal{B}(\mathcal{O})^n$ represent votes cast by the agents. An element $\vec{v} \in \mathcal{B}(\mathcal{O})^n$ is called a *vote profile*. A tallying function $\mathsf{Ag} : \mathcal{B}(\mathcal{O})^n \to \mathcal{O}$ maps vote profiles to candidates.

Definition 20 *Let $\mathcal{A}$ be a set of n agents and $\mathcal{O}$ a set of m candidates. A* **voting procedure** *is a pair $\mathcal{V} = \langle \mathcal{B}(\mathcal{O}), \mathsf{Ag} \rangle$, where $\mathcal{B}(\mathcal{O})$ is a set of ballots and $\mathsf{Ag} : \mathcal{B}(\mathcal{O})^n \to \mathcal{O}$ is a* **tallying function**, *or a* **scoring function**.

The following are examples of some well-known voting procedures. Let $\mathcal{A}$ be a set of n agents and $\mathcal{O}$ a set of m candidates.

[3][9] shows that the Gibbard-Satterthwaite theorem holds when ties are permitted.

Plurality Voting: The voting procedure $\mathcal{V}_P = \langle \mathcal{B}(\mathcal{O}), \mathsf{Ag} \rangle$ is called plurality voting if $\mathcal{B}(\mathcal{O}) = \{\{o\} \mid o \in \mathcal{O}\}$ and Ag selects the candidate with the largest number of votes. For simplicity, in the case of ties, we assume that Ag randomly selects among the candidates with the most votes. We assume this throughout the chapter.

Approval Voting: The voting procedure $\mathcal{V}_A = \langle \mathcal{B}(\mathcal{O}), \mathsf{Ag} \rangle$ is called approval voting if $\mathcal{B}(\mathcal{O}) = 2^{\mathcal{O}}$ and Ag selects a candidate with the largest number of approvals.

Borda Count: The voting procedure $\mathcal{V}_B = \langle \mathcal{B}(\mathcal{O}), \mathsf{Ag} \rangle$ is called Borda count if $\mathcal{B}(\mathcal{O}) =$ **Pref**, i.e., ballots are linear orderings of $\mathcal{O}$. The scoring function Ag is slightly more complicated then above. Each candidate ranked highest by a voter receives the most points, the next-highest receives the next-most points, and so on. Then Ag selects the candidate with the largest point total. When there are m candidates, then the usual Borda points are $m-1, m-2, \ldots, 0$ for the first choice, second choice, ..., last choice.

Hare System: The voting procedure $\mathcal{V}_H = \langle \mathcal{B}(\mathcal{O}), \mathsf{Ag} \rangle$ is called the Hare system, or single transferable vote, if $\mathcal{B}(\mathcal{O}) =$ **Pref** and Ag works as follows. If no candidate receives a majority of first-place votes, then the candidate with the fewest first-place votes is dropped and his second place votes are given to the remaining candidates. This elimination process continues until one candidate receives a simple majority.

Given a voting procedure $\mathcal{V}$ and an agent i's preference P_i, we can ask if a vote $v \in \mathcal{B}(\mathcal{O})$ is a "sincere" representation of P_i. For some voting procedures there is an objective answer to this question. For example, if we assume that the voting procedure is $\mathcal{V}_P$, then a vote v is sincere with respect to preference P iff v is the maximal[4] element of P. However, for some voting procedures, such as approval voting, more information is needed to determine whether or not a vote is a sincere reflection of a preference P. In approval voting, whether a vote v (which is a subset of $\mathcal{O}$) is sincere depends on both a preference P *and* where the agent places its cut-off point between approved and 'dis-approved' candidates.

In order to capture the above notion of a "sincere vote", we assume for each agent i a function S_i, called the **sincere vote** function, between the set of preferences **Pref** and the set of subsets of ballots. I.e., $S_i : \mathbf{Pref} \rightarrow 2^{\mathcal{B}(\mathcal{O})}$. Typically, we will assume that for each $P \in \mathbf{Pref}$, $S_i(P)$ is a singleton, but this is not necessary. If i's preference is P_i and $v \in S_i(P_i)$, then v is said to be a **sincere** vote corresponding to P_i. The voter i is said to **strategize** with respect to a preference P if i selects a vote v that is not in the set $S_i(P)$. See [18] for a definition of a "sincere vote" in a variety of context and [80] for a discussion of similar issues.

Assume that the agents' true preferences are $\vec{\mathcal{P}}^* = (P_1^*, \ldots, P_n^*)$ and *fixed* for the

[4]Recall that we are assuming preferences are linear orders.

remaining discussion. Given a vote profile $\vec{v}$ of *actual* votes, we ask whether agent i will change its vote if given another chance to vote. Let $\vec{v}_{-i}$ be the vector of all *other* agents' votes. Then given $\vec{v}_{-i}$ and i's true preference P_i^*, there will be a (nonempty) set X_i of votes that are i's *best response* to $\vec{v}_{-i}$. Of course, whether v is a best response for agent i to $\vec{v}_{-i}$ will depend on the voting procedure. We will be more specific below about what constitutes a best response to a vector $\vec{v}_{-i}$ for agent i.

Suppose that for each $i \in \mathcal{A}$, f_i selects selects one such best response from X_i. We assume that agent's will only strategize if necessary. That is, if $v \in S_i(P_i^*)$ and $v \in X_i$, then f_i will select v(if more than one such v exists, then let f_i select one of these votes). Let $f(\vec{v}) = (f_1(\vec{v}_{-1}), \ldots, f_n(\vec{v}_{-n}))$. We call f a **strategizing function**. If $\vec{v}$ is a fixed point of f (i.e., $f(\vec{v}) = \vec{v}$), then $\vec{v}$ is a *stable* outcome. We define f^n recursively by $f^1(\vec{v}) = f(\vec{v})$, $f^n = f(f^{n-1}(\vec{v}))$, and say that f is **stable at level** n if $f^n(\vec{v})) = f^{n-1}(\vec{v})$. It is clear that if f is stable at level n, then f is stable at all levels m where $m \geq n$. Also, if the initial votes of the $\vec{v}$ are a fixed point of f then all levels are stable.

The following two examples demonstrate the type of situations that we have in mind. The first example is taken from [16].

Example 5.2.1 (Brams and Fishburn [16]) *Suppose that there are four candidates* $\mathcal{O} = \{o_1, o_2, o_3\}$ *and nine voters divided into three groups:* A, B *and* C. *Suppose that the sizes of the groups are given as follows:* $|A| = 4$, $|B| = 3$, *and* $|C| = 2$. *We assume that all the agents in each group have the same true preference and that they all vote the*

Condorcet candidate, *i.e., a candidate who defeats every other candidate in a pairwise contest.*

Brams and Fishburn go on to generalize this example and show that if the agents follow the protocol described above, then under plurality voting, if the Condorcet candidate is not one of the top two candidates identified by the poll, then that Condorcet candidate will always lose. In the above example, the protocol is set up so that the second round of votes is a fixed point, i.e., the voters will not change their votes a second time. The next example describes a situation in which a fixed point does not occur until round IV. The following example was first presented in [23].

Example 5.2.2 *Suppose that there are four candidates* $\mathcal{O} = \{o_1, o_2, o_3, o_4\}$ *and five groups of voters:* A, B, C, D *and* E. *Suppose that the sizes of the groups are given as follows:* $|A| = 40$, $|B| = 30$, $|C| = 15$, $|D| = 8$ *and* $|E| = 7$. *We assume that all the agents in each group have the same true preference and that they all vote the same way. Suppose that the voting procedure is plurality voting* ($\mathcal{V}_P$). *Hence for each* $i \in \mathcal{A}$, $v \in S_i(P_i)$ *iff* v *is the maximal element of* P_i. *The agents' true preferences are as follows:*

$$P_A^* = o_1 >_{P_A^*} o_4 >_{P_A^*} o_2 >_{P_A^*} o_3$$

$$P_B^* = o_2 >_{P_B^*} o_1 >_{P_B^*} o_3 >_{P_B^*} o_4$$

$$P_C^* = o_3 >_{P_C^*} o_2 >_{P_C^*} o_4 >_{P_C^*} o_1$$

$$P_D^* = o_4 >_{P_D^*} o_1 >_{P_D^*} o_2 >_{P_D^*} o_3$$

$$P_E^* = o_3 >_{P_E^*} o_1 >_{P_E^*} o_2 >_{P_E^*} o_4$$

We assume that the agents all use the following protocol. If the current winner is o, then agent i will switch its vote to some candidate o' provided:

1. *i prefers o' to o, formally* $o' >_{P_i} o$, *and*
2. *the current total for o' plus agent i's group's votes for o' is greater than the current total for o.*

By this protocol an agent (thinking only one step ahead) will only switch its vote to a candidate which is currently not the winner. Initially, we assume that the agents all report their (unique) sincere vote. The following table describes what happens if the agents use this protocol. The candidates in bold are the winner of the current election round.

Size	*Group*	*I*	*II*	*III*	*IV*
40	*A*	$\mathbf{o_1}$	o_1	o_4	$\mathbf{o_1}$
30	*B*	o_2	$\mathbf{o_2}$	$\mathbf{o_2}$	o_2
15	*C*	o_3	$\mathbf{o_2}$	$\mathbf{o_2}$	o_2
8	*D*	o_4	o_4	o_1	o_4
7	*E*	o_3	o_3	o_1	$\mathbf{o_1}$

In round I, everyone reports their top choice and o_1 *is the winner.* C *likes* o_2 *better than* o_1 *and its own total plus* B*'s votes for* o_2 *exceed the current votes for* o_1*. Hence by the protocol,* C *will change its vote to* o_2*.* A *will not change its vote in round II since its top choice is the winner.* D *and* E *also remain fixed since they do not have an alternative like* o' *required by the protocol. In round III, group* A *changes its vote to* o_4 *since it is preferred to the current winner* (o_2) *and its own votes plus* D*'s current votes for* o_4 *exceed the current votes for* o_2*.* B *and* C *do not change their votes. For* B*'s top choice* o_2 *is the current winner and as for* C*, they have no* o' *better than* o_2 *which satisfies condition 2). Ironically, Group* D *and* E *change their votes to* o_1 *since it is preferred to the current winner is* o_2 *and group* A *is currently voting for* o_1*. Finally, in round IV, group* A *notices that* E *is voting for* o_1 *which* A *prefers to* o_4 *and so changes its votes back to* o_1*. The situation stabilizes with* o_1 *which, as it happens, is also the Condorcet winner. I.e., it is easy to check that by following the above protocol,* $f((o_1, o_2, o_2, o_4, o_1)) = (o_1, o_2, o_2, o_4, o_1)$*. Thus* f *stabilizes at stage 4.*

We are now in a position to be more specific about what constitutes a best response for an agent. In the above examples, the agents' decisions to strategize were based on a predefined protocol. Note that in both examples, the agents behaved myopically, that is the protocol only took into account information from the current round. This restriction can be relaxed to allow the agents to make decisions based on, for example, all previous rounds. The important point from both examples is that the agents' voting strategies were explained by assuming that the agents are all following a particular protocol. In general, the agents may not necessarily all follow the same protocol as we have assumed in the above example. We will now discuss some issues relevant to formalizing this notion notion of a protocol.

In general there may be a lot of reasons why an agent may decide to change its vote. Of course an agent may change its vote *because* its preferences have changed. However, this is not the phenomenon we are trying to capture in this chapter. We are interested in situations in which each agent's preference is fixed and the agent is trying to decide which vote best reflects its preference given the current situation. We assume that an agent's decision to change its vote will be based on three pieces of information. The first is the agent's actual preference. The second is information about the current vote profile, called **poll** information. The third is information about the number of agents that have the same preference.

In the above example, we assumed that during each round the agents were told

the total number of votes each candidate received. In general, the form of the polling information will depend on the voting protocol that is being used. For example, knowing the total number of votes each agent received will be relevant for any voting procedure that selects the winner based solely on the total number of votes the candidates receive; however, this information will be less useful when the voting method is Borda count. In the interest of concreteness, we will restrict attention to voting methods, such as approval voting or plurality voting, that select the winner based solely on the total number of votes that the candidate receives. Thus we can model the poll information as a function $\pi : \mathcal{B}(\mathcal{O}) \to \mathbb{N}$, where $\pi(v) = n$ is interpreted as n voters have selected voter v. Let Π be the set of all such functions, i.e., the set of polls. For each poll π, let $W(\pi)$ be the candidate that would win (according to Ag) the election if the agents vote according[6] to π. Finally, we note that each voting profile induces a poll. That is if $\vec{v}$ is a voting profile, then we can define a poll $\pi_{\vec{v}}$ as follows, for each $v \in \mathcal{B}(\mathcal{O})$, $\pi_{\vec{v}}(v) = \sum_{v_i = v} 1$.

The second piece of information that an agent i uses to decide whether or not to change its vote is the size of the group of agents that share i's true preference. Typically, a single agent changing its votes will not affect the outcome of an election. However, as in the above example, agents will change their vote in part *because* they assume that they are part of a group which has enough weight to swing an election. In the

[6]Of course, a poll does not list which agent voted for which candidate; however a winner can still be determined provided we assume that the Ag function is invariant under permutation of voters. This certainly true of many voting procedures.

above example, this number was constant for each agent. That is we assumed that each agent knew the exact size of the set of agents that share its actual preference. However in general, this information may not be known to an agent or the agent may only have partial information about the size of the number of agents that share its actual preference. This will be modeled by a group size function γ_i from a finite sequence of polls to the natural numbers. That is $\gamma_i : \Pi^* \to \mathbb{N}$ where $\gamma_i(\pi_1 \cdots \pi_k) = l$ means that after the series of polls $\pi_1, \ldots, \pi_n$ agent i believes that there are l agents that have the same actual preference as itself. Let Γ_i be the set of all possible such functions.

We are now in a position to formally define a protocol for an agent i. By an **election** we mean a sequence of polls, i.e., an element of Π^*. A **protocol** for agent i is a function $\Delta_i : \mathbf{Pref} \times \Pi^* \times \Gamma_i \to \mathcal{B}(\mathcal{O})$. Thus if $\sigma \in \Pi^*$ is an election and $\gamma_i \in \Gamma_i$ is a group function, then $\Delta_i(P, \sigma, \gamma_i) = v'$ means that agent i will use ballot v' in the next poll. Notice that we are assuming that the agent's use the entire election when making its decision to change its vote. This is assumed in the interest of generality. In other words, we assume that all agents have access to the election information, but whether or not they use all of that information is another issue all together. Call the vector $\vec{\Delta}$ a *group protocol.* We assume that in the absence of information, agents will vote according to their actual preferences. That is if σ is the empty string, then for each $i \in \mathcal{A}$, $\Delta_i(P_i^*, \sigma, \gamma_i) = v \in S_i(P_i^*)$.

Given a group protocol, we say that a strategizing function f is generated by $\vec{\Delta}$,

written $f_{\vec{\Delta}}$ if $f_{\vec{\Delta}}$ is defined as follows. Suppose that $\sigma = \pi_1 \cdots \pi_k$ is the current poll information and $\vec{v}$ is the current vote profile. Then we define

$$f_{\vec{\Delta}}(\vec{v}) = (\Delta_1(P_1^*, \sigma, \gamma_n(\sigma)), \ldots, \Delta_n(P_n^*, \sigma, \gamma_n(\sigma)))$$

Returning to our example. We will now demonstrate how to formalize our second example using the above machinery. First of all we assume that the agents all knew the exact number of agents that share their actual preferences. Thus for each $i \in \mathcal{A}$, γ_i is the constant function described in the example above. For example, for each election σ, $\gamma_i(\sigma) = 40$ iff $i \in A$. The Δ_i functions can be described as follows. In order to ease exposition, we will identify singleton subsets with their element. In other words, we are assuming that $\mathcal{B}(\mathcal{O}) = \mathcal{O}$. Let $W_2(\pi)$ denote the candidate that receives the second highest number of votes. The protocol that each agent follows in the first example can be described as follows. First of all we need some notation: for each pair of candidate $o, o' \in \mathcal{O}$, let $C_{P_i}(o, o')$ choose that candidate preferred according to P_i.

$$\Delta_i^1(P_i^*, \sigma, \gamma_i) = \begin{cases} o' & (v \neq W(\pi) \text{ or } v \neq W_2(\pi)) \text{ and } o' = C_{P_i^*}(W(\pi), W_2(\pi)) \\ v & \text{otherwise} \end{cases}$$

where v is the current vote (similarly for the next example). The second example can

be formalized as follows:

$$\Delta_i^2(P_i^*, \sigma, \gamma_i) = \begin{cases} o' & P_i^*(o', W(\mathsf{last}(\sigma))) \text{ and } \gamma_i(\sigma) + \mathsf{last}(\sigma)(o') > \mathsf{last}(\sigma)\pi(o) \\ v & \text{otherwise} \end{cases}$$

Notice that in Δ_i^1, the group function is not used in the definition. In other words, in Example 5.2.1, the agents need not have any information about the size of the group that share their preferences in order to follow the protocol. Whereas, in the second example, the size of the group plays a key role in the agent's decision to change its current vote. Putting everything together, we can now define a voting model.

Definition 21 (Voting Model) *Given a set of n agents $\mathcal{A}$ and m candidates $\mathcal{O}$, a* **voting model** *is a tuple $\langle \mathcal{V}, \vec{\mathcal{P}}^*, \{S_i\}_{i\in\mathcal{A}}, f\rangle$ where $\mathcal{V}$ is a voting procedure, each S_i is a sincere vote function for agent i; and f is a strategizing function. We say f is generated by a group protocol $\vec{\Delta}$ if $f = f_{\vec{\Delta}}$.*

We now point out that if there are more than two candidates, then for every voting procedure $\mathcal{V}$, there *must* be instances in which the generated strategizing function f never stabilizes:

Theorem 5.2.1 *If $\mathcal{O}$ has three or more candidates, then for any given voting procedure $\mathcal{V}$, there exists an initial vector of preferences such that f never stabilizes.*

This follows easily from the Gibbard-Satterthwaite theorem. Suppose not, then we show that there is a strategy-proof tallying function contradicting the Gibbard-Satterthwaite theorem. Suppose that Ag is an arbitrary tallying function and $\vec{\mathcal{P}}^*$ the vector of true preferences. Suppose there always is a level k at which f stabilizes given the agents' true preferences $\vec{\mathcal{P}}^*$. But then define Ag$'$ to be the outcome of applying Ag to $f^k(\vec{\mathcal{P}}^*)$ where $\vec{\mathcal{P}}^*$ are the agents' true preferences. Then given some obvious conditions on the strategizing function f, Ag$'$ will be a strategy-proof tallying function contradicting the Gibbard-Satterthwaite theorem. Hence there *must be* situations in which f never stabilizes.

Since our candidate and agent sets are finite, if f does not stabilize then f **cycles.** We say that f has a cycle of length n if there are n different votes $\vec{\mathcal{P}}_1, \ldots \vec{\mathcal{P}}_n$ such that $f(\vec{\mathcal{P}}_i) = \vec{\mathcal{P}}_{i+1}$ for all $1 \leq i \leq n-1$ and $f(\vec{\mathcal{P}}_n) = \vec{\mathcal{P}}_1$.

The following is an example of a situation in which the associated strategizing function never stabilizes:

Example 5.2.3 *Consider three candidates* $\{o_1, o_2, o_3\}$ *and 100 agents. Suppose that their are three groups of agents A, B and C. The size of each group is* $|A| = 40$, $|B| = 30$ *and* $|C| = 30$. *The actual preferences are given as follows:*

$$P_A^* = o_1 >_{P_A^*} o_2 >_{P_A^*} o_3$$

$$P_B^* = o_2 >_{P_B^*} o_3 >_{P_B^*} o_1$$

$$P_C^* = o_3 >_{P_C^*} o_1 >_{P_C^*} o_2$$

Assume that the agents use the following protocol. An agent i will switch its vote for o to o′ provided (assume w is the current winner)

1. *o′ is i's second choice and the current winner is i's last choice, or*

2. *o′ is i's top choice and the current winner is i's top choice.*

Assuming that the voting protocol is plurality voting and that all agents follow the above protocol generates the following table.

Size	*Group*	*I*	*II*	*III*	*IV*	*V*	*VI*	*VII*	*VIII*	*IX*	···
40	*A*	**o_1**	o_1	o_2	**o_2**	**o_2**	**o_1**	**o_1**	o_2	o_1	···
30	*B*	o_2	**o_3**	**o_3**	**o_2**	**o_2**	o_2	o_3	**o_3**	**o_3**	···
30	*C*	o_3	**o_3**	**o_3**	o_3	o_1	**o_1**	**o_1**	**o_3**	**o_3**	···

After reporting their initial preferences, candidate o_1 will be the winner with 40 votes. The members of group B dislike o_1 the most, and will strategize in the next election by

reporting o_3 as their preference. So, in the second round, o_3 will win. But now, members of group A will report o_2 as their preference, in an attempt to draw support away from their lowest ranked candidate. o_3 will still win the third election, but by changing their preferences (and making them public) group A sends a signal to group B that it should report its true preference - this will enable group A to have its second preferred candidate o_2 come out winner. This cycling will continue indefinitely; o_2 will win for two rounds, then o_1 for two rounds, then o_3 for two, etc.

5.3 Conclusion and Further Work

Our results suggest that election-year opinion polls are a way to effectively turn a one-shot game, i.e., an election, into a many-round game that may induce agents to strategize. Opinion polls make voters' preferences public in an election year and help voters decide on their strategies on the day of the election. For the rest of the paper, we will refer to opinion polls also as elections.

We have explored some properties of strategic voting and noted that the Gibbard-Satterthwaite theorem only applies in those situations where agents can obtain the appropriate knowledge. In example 5.2.2 the Condorcet winner - the winner in pairwise head-to-head contests - was picked via strategizing. Since our framework makes it possible to view opinion polls as the $n-1$ stages of an n-stage election, it implies that communication of voters' preferences and the results of opinion polls can play an

important role in ensuring rational outcomes to elections. Put another way, while the Gibbard-Satterthwaite theorem implies that we are stuck with voting mechanisms susceptible to strategizing, our work indicates ways for voters to avoid irrational outcomes using such mechanisms.

Chapter 6

Conclusions

Social software brings together ideas from philosophical logic, game theory and computer science. As is true in many interdisciplinary fields, an important component of social software research is developing a "common language" in which experts from the different fields with diverse backgrounds can compare and contrast their results. The logical systems developed in this thesis are a step in this direction. In particular the frameworks discussed in Chapters 2, 3 and 4 are intended to represent types of situation that are of interest to both game theorists and computer scientists. We briefly summarize the results presented in this thesis.

Chapter 2 introduces the basic formal framework used to represent social interactive situations. We view a social interactive situation as consisting of a collection of sequences of "events" (called histories), where the exact interpretation of an event depends on the

application. For example, since we where modeling communication among agents in Chapter 4, an event in Chapter 4 consisted of a (one-way) communication between two agents. Intuitively, each global history (infinite sequence of events) is a possible way the situation could have evolved. At any moment $t \in \mathbb{N}$ there is a finite history and a possibly infinite future. Some of the events are caused by an agent, i.e., an agent can perform a particular action and others are caused by nature (which can be viewed as a special type of agent). In Chapter 2, we show how to start from this basic framework to construct models that have been used by computer scientists (history based knowledge models) to study distributed algorithms and models that have been used by game theorists (extensive games) to study multi-agent social situations.

Chapters 3 and 4 are the main contributions of this thesis. The contribution of Chapter 3 is primarily conceptual. A formal framework for reasoning about actions, knowledge and obligations is described. This framework is shown to naturally capture our intuitions about various deontic dilemmas. Chapter 4 takes on the task of finding a model for multi-agent knowledge and communication. A logical system and semantics is defined (based on the history based structures from Chapter 2) which is shown to be decidable. The analysis in Chapter 5 suggests that a certain level of knowledge is required in order to make the Gibbard-Sattertwaithe Theorem effective.

In Chapter 1, we discussed the three main areas of social software research. This thesis focused on the first of these three areas: mathematical models of social situations.

Of course there are a number of issues relevant for developing mathematical models of social situations which we have glossed over or not discussed at all. We point to two of the most relevant and pressing issues.

Logical Omniscience

We have argued that an important part of any formal theory of social procedures, is how knowledge of the individual agents is represented. Of course, without a certain amount of idealization, the mathematics may become too complicated to be of any practical use. As such, we must learn to live with the fact that there may be unrealistic assumptions made about the agents in our mathematical models. One such assumption, called logical omniscience, has been discussed by a number of different authors ([70, 98, 99, 3, 37, 34]). Among other things, logical omniscience implies that an agent knows all the logical consequences of its knowledge (see [70] for a definition of logical omniscience). Perhaps we can live with such an unrealistic assumption. However, as Stalnaker astutely points out, "Any context where an agent engages in reasoning is a context that is distorted by the assumption of deductive omniscience, since reasoning (at least deductive reasoning) is an activity that deductively omniscient agents have no use for. Deliberation, to the extent that it is thought of as a rational process of figuring out what one should do given one's priorities and expectations is an activity that is

unnecessary for the deductively omniscient." [98]. This raises some very important questions for the social software scientist. In particular, what exactly is the role that epistemic logic plays in the analysis of social procedures? If we hope to use epistemic logic to show that in a given situation each agent has enough information in order to follow the rules of some social procedure, then clearly the assumption of deductive closure is much too strong.

We briefly comment on some solutions to the logical omniscience problem that have been offered in the literature. The first is to consider logical systems without the K axiom $(K_i(\phi \rightarrow \psi) \rightarrow K_i\phi \rightarrow K_\psi)$. These so-called classical systems of modal logic are interpreted in *neighborhood models*, or Scott-Montague models. Essentially the idea is to shift from thinking of knowledge as being derived from an inaccessibility relation a la Hintikka to being explicitly described as part of the model, i.e. at each world the neighborhood function returns the set of propositions that are known at that world. The textbook [21] has a discussion on these models for propositional modal logic and [1] has a discussion of these models in the context of first-order modal logic. Perhaps the most promising solution to the logical omniscience problem relevant for the analysis of social procedures was offered by Rohit Parikh in [70]. Parikh introduces the notion of *behavioral knowledge*. Say that agent i *b-knows* a formula ϕ if there are three mutually incompatibly actions a, b, c such that i does a only if ϕ is true, does b only if ϕ is false, and i has just done a (there are no restrictions on c). The definition of knowledge in

Chapter 2 for history based frames has some of the flavor of behavioral knowledge. In history based frames, knowledge is derived from the agent's observation of the events that have taken place which in turn is based on the behavour of the agents. However, much more can be said, but this is a topic for future research. Finally, Fitting and Artemov have recently been pursuing a different line of reasoning. The main idea is to replace formulas of the form $K_i\phi$ with formulas of the form $t : \phi$, which are intended to mean that t is a "justification", or provides "evidence", for ϕ. This idea of making modalities explicit was introduced by Artemov in [4] to provide a logic of explicit proofs. See [37, 38, 3] for more on this topic.

Empirical Studies

It is well-known that human agents do not necessarily behave as implied by the models we have described in this thesis. To what extent these models have captured social interactions among human agents is still very much an open problem. Solving this problem requires collaboration with psychologists to design experiments that can test our hypotheses. In fact, there are a number of experiments that have already been conducted that may be of interest to social software researchers. As an example, we discuss one experiment by MIT psychologist Alex Bavelas.

In the 1950s, Alex Bavelas ran a series of experiments on the effects of group structure

and communication on task performance - with surprising results. In one of them, which he described to the Cybernetics group at the Interdisciplinary Macy conferences [33, 8] held in New York City between 1946 and 1953, participants were asked to pick a number from the range 0-5, to write it down on a piece of paper and to give it to the experiment moderator. The five participants in the group were supposed to generate guesses so that their total added up to 17. Participants were not allowed to communicate with each other and were not told what the other participants had guessed. The experiments were conducted in two different ways. In one, participants were told whether their guesses had resulted in success or not, without telling them what their total was. So, if in fact the numbers did not add up to 17, the participants were told 'sorry, try again' and had to guess again. In the other kind of experiment, the participants were told whether their guess undershot or overshot i.e., whether they had guessed too high or too low (the actual total was also announced). The second experiment then, gave the participants 'more information' about their combined guess.

Bavelas reported that the groups always took longer to converge on the correct answer when playing the second, 'more informative', kind of game. Prima facie these results are surprising; the second kind of game is ostensibly more informative; an announcement has been made, which is public and the contents of which are common knowledge. Why then do the players take longer to arrive at a correct combined guess in the second form of the game? Note that if a single agent is trying to guess a particular number, then

of course the more informative response of the referee will cause the agent to guess the number at a faster pace.

The setup of the Bavelas experiments is reminiscent of many of the epistemic puzzles which have recently been the focus of much scrutiny. In particular, the well-known muddy children puzzle[1] shares a number of features with the Bavelas experiments. In both situations a group of agents are reacting to public announcements. However, in the case of the muddy children the reaction involves an *epistemic update* whereas in the Bavelas experiment the reaction involves a 'co-ordinated' response by the group. The difference in the two situations is in what we find puzzling. In the muddy children puzzle, we are surprised that the repeated announcement of seemingly 'useless' information ('some child has mud on its forehead' and 'we don't know if we have mud on our forehead') actually increases the agents' information. That is, the announcements in the muddy children puzzle - while seemingly not conveying information - reduce the size of the uncertainty space. In the Bavelas experiments, the announcements - while seemingly reducing the size of the uncertainty space - inhibit the performance of the group on the task at hand. An initial analysis of this experiment is offered in [22].

Finally, we note that the reader may feel somewhat unsatisfied with the conclusions drawn in this chapter. We seem to have raised a number of serious doubts about the applicability of our models. In order to put the reader's mind at ease, we let Johann

[1]See [34] for a discussion of the formal properties of the puzzle.

Von Neumann have the last word. This is particularly fitting as Von Neumann was a very influential figure in both the development of computer science and game theory.

> The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work. — Johann Von Neumann

Bibliography

[1] Horacio Arló-Costa and Eric Pacuit. Classical systems of first-order modal logic. Technical report, Carnegie Mellon University, 2004.

[2] K. J. Arrow. *Social choice and individual values (2nd edition).* Wiley, New York, 1963.

[3] S. Artemov and E. Nogina. Basic epistemic logics with justifications. Technical report, CUNY Graduate Center, 2004.

[4] Sergei Artemov. Logic of proofs. *Annals of Pure and Applied Logic*, 67:29–59, 1994.

[5] R. Aumann. Interactive epistemology I: Knowledge. *International Journal of Game Theory*, 28:263–300, 1999.

[6] M. O. L. Bacharach. The epistemic structure of a theory of a game. In M.O.L. Bacharach, L.A. Gerard-Varet, P. Mongin, and H.S. Shin, editors, *Epistemic Logic*

and the Theory of Games and Decisions, pages 303 – 344. Kluwer Academic Publishers, 1997.

[7] Alexandrau Baltag and Larry Moss. Logics for epistemic programs. *Synthese: Knowledge, Rationality, and Action*, 2:165 – 224, 2004.

[8] Alex Bavelas. Communication patterns in problem-solving groups. In Claude Pias, editor, *Cybernetics — Kybernetik. The Macy-Conferences 1946-1953: Volume 1 Transactions/Protokolle.* Diaphanes, 2003.

[9] Jean-Pierre Benoit. Strategic manipulation in games when lotteries and ties are permitted. *Journal of Economic Theory*, 102:421–436, 2002.

[10] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, Cambridge, 2002.

[11] G. Bonanno and P. Battigalli. Recent results on belief, knowledge and the epistemic foundations of game theory. *Research in Economics*, 53(2):149–225, June 1999.

[12] G. Bonanno and K. Nehring. Agreeing to disagree: a survey. Document prepared of an invited lecture at the Workshop on Bounded Rationality and Economic Modelling, July 1997.

[13] Giacomo Bonanno. A characterization of von neumann games in terms of memory. *Synthese*, 139(2):237–256, March 2004.

[14] S. J. Brams and A. D. Taylor. *Fair Division: From cake-cutting to dispute resolution.* Cambridge University Press, Cambridge, 1996.

[15] S. J. Brams and A. D. Taylor. *The Win-Win Solution.* W. W. Norton and Company, New York, 1999.

[16] Steven Brams and Peter Fishburn. *Approval Voting.* Birkhauser, Boston, 1983.

[17] Steven J. Brams. Voting Procedures. In *Handbook of Game Theory*, volume 2, pages 1055–1089. Elsevier, 1994.

[18] Steven J. Brams and Peter C. Fishburn. Voting Procedures. In *Handbook of Social Choice and Welfare.* North-Holland, 1994.

[19] Adam Brandenburger and H.J. Keisler. An impossibility theorem on belies in games. available at http://pages.stern.nyu.edu/ abranden/itbg072904.pdf, July 2004.

[20] S. Callander. Bandwagons and momentum in sequential voting. Working Paper: available at http://www.kellogg.northwestern.edu/faculty/callander/, 2003.

[21] Brian Chellas. *Modal Logic: An Introduction.* Cambridge University Press, Cambridge, 1980.

[22] Samir Chopra and Eric Pacuit. When more is less: The bavelas experiments. Working Paper, 2005.

[23] Samir Chopra, Eric Pacuit, and Rohit Parikh. Knowledge-theoretic properties of strategic voting. In Jose Julio Alferes and Joao Leite, editors, *Proceedings of JELIA 2004*, Lecture Notes in Artificial Intelligence, pages 18–30. Springer, 2004.

[24] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking.* MIT Press, Boston, 1999.

[25] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? In *Proceedings of the Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2003.

[26] V. Conitzer and T. Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2002.

[27] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the International Joint Conference on Artificial Intelligences (IJCAI)*. Morgan Kaurmann, 2003.

[28] Lorrie Faith Cranor. *Declared-Strategy Voting: An Instrument for Group Decision-Making.* PhD thesis, Washington University Sever Institute of Technonology, 1996.

[29] Andrew Dabrowski, Larry Moss, and Rohit Parikh. Topolgical reasoning and the logic of knowledge. *Annals of Pure and Applied Logic*, 78:73 – 110, 1996.

[30] R. K. Dash, D. Parkes, and N. R. Jennings. Computational mechanism design : A call to arms. *IEEE Intelligent Systems*, 18(6):40–47, 2003.

[31] D. Davidson. *Essays on actions and events.* Oxford University Press, New York, 1980.

[32] E. Dekel and M. Piccione. Sequential voting procedures in symmetric binary elections. *Jouranl of Political Economy*, 2000.

[33] Jean-Pierre Dupuy. *The Mechanization of Mind.* Princeton University Press, Princeton, 2000.

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

[35] Melvin Fitting. Many-valued modal logics. *Fundam. Inform.*, 15(3-4):235–254, 1991.

[36] Melvin Fitting. Many-valued modal logics II. *Fundam. Inform.*, 17(1-2):55 – 73, 1992.

[37] Melvin Fitting. The logic of proofs, semantically. *Annals of Pure and Applied Logic*, 132:1 – 25, 2005.

[38] Melvin Fitting. The logic of explicit knoweldge. In *Logica 2004 Proceedings*, to appear.

[39] R. W. Floyd. Assigning meanings to programs. *Proc. Symp. Appl. Math*, 19:19–31, 1967.

[40] T. French, R. van der Meyden, and M. Reynolds. Axioms for logics of knowledge and past time: Synchrony and unique initial states. appeared in the pre-proceedings of the Conference on Advances in Modal Logic, Manchester, Sept 2004.

[41] Konstantinos Georgatos. *Modal Logics for Topological Spaces.* PhD thesis, CUNY Graduate Center, 1993.

[42] Konstantinos Georgatos. Knowledge theoretic properties of topological spaces. In M. Masuch and L. Polos, editors, *Lecture Notes in Artificial Intelligence*, volume 808, pages 147–159. Springer-Verlag, 1994.

[43] Konstatinos Georgatos. Knowledge on treelike spaces. *Studia Logica*, 1997.

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

[45] Allan Gibbard. Manipulation of Voting Schemes: A General Result. *Econometrica*, 41(4):587–601, 1973.

[46] J. Glazer and C. A. Ma. Efficient allocation of a 'prize' — king solomon's dilemma. *Games and Economic Behavior*, 1:222 – 233, 1989.

[47] P. Gochet and P. Gribomont. Epistemic logic. Forthcoming in *The Handbook of History and Philosophy of Logic*, Elsevier, edited by D. Gabbay and J. Woods.

[48] Valentin Goranko. Temporal logics of computation. Notes prepared for a course at the 12th European Summer School in Logic, Language and Information (http://general.rau.ac.za/maths/goranko/new/papers/esslli2000.pdf).

[49] Adam Grove. Two modellings for theory of change. *Journal of Philosophical Logic*, 17:157 – 179, 1988.

[50] Joseph Halpern. A computer scientist looks at game theory. *Games and Economic Behavior*, 45(1):114 – 131, 2003.

[51] Joseph Halpern, Ron van der Meyden, and Moshe Vardi. Complete axiomatizations for reasoning about knowledge and time. *SIAM Journal of Computing*, 33(2):674 – 703, 2004.

[52] Joseph Halpern and Moshe Vardi. The complexity of reasoning about knowledge and time. *J. Computer and System Sciences*, 38:195 – 237, 1989.

[53] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, Boston, 2000.

[54] Bernhard Heinemann. Temporal aspects of the modal logic of subset spaces. *Theoretical Computer Science*, 224(1-2):135 – 155, 1999.

[55] Bernhard Heinemann. Extending topological nexttime logic. In S. D. Goodwin and A. Trudel, editors, *Temporal Representation and Reasoning*, pages 87–94. IEEE Computer Society Press, 2000.

[56] Bernhard Heinemann. A hybrid logic of knowledge supporting topological reasoning. In *Algebraic Methodology and Software Technology, AMAST 2004*, Lecture Notes in Computer Science. Springer, 2004.

[57] R. Hilpinen. Deontic logic. In Lou Goble, editor, *Blackwell guide to philosophical logic*, pages 159–182. Blackwell, 2001.

[58] Jaakko Hintikka. *Knowledge and Belief: An Introduction to the Logic of the Two Notions.* Cornell University Press, Ithaca, 1962.

[59] C. A. R. Hoare. An axiomatic basis for computer programming. *Comm. Assoc. Comput. Mach.*, 12, 1969.

[60] John Horty. *Agency and Deontic Logic.* Oxford, Oxford, 2001.

[61] Barteld Kooi. *Knowledge, Chance and Change.* PhD thesis, University of Groningen, 2003.

[62] Barteld Kooi and Hans van Ditmarsch. The secret of my success. *Synthese*, forthcoming.

[63] Alessio Lomuscio and Marek Sergot. Deontic interpreted systems. *Studia Logica*, 75:63 – 92, 2003.

[64] Larry Moss and Rohit Parikh. Topological reasoning and the logic of knowledge. In Yoram Moses, editor, *Proceedings of TARK IV*. Morgan Kaufmann, 1992.

[65] M. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, Boston, 1994.

[66] Eric Pacuit and Rohit Parikh. The logic of communication graphs. In J. Leita, A. Omicini, P. Torroni, and P. Yolum, editors, *Proceedings of DALT 2004*, number 3476 in Lecture Notes in AI, pages 256 – 269. Springer, 2005.

[67] Eric Pacuit and Samer Salame. Majority logic. In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 598–605. AAAI Press, June 2 - 5 2004.

[68] Palfrey and Rosenthal. A strategic calculus of voting. *Public Choice*, 41:7 – 53, 1983.

[69] Palfrey and Rosenthal. Voter participation and strategic uncertainty. *American Political Science Review*, 41:62 – 78, 1985.

[70] R. Parikh. Knowledge and the problem of logical omniscience. In Z. Ras and M. Zemankova, editors, *ISMIS-87*, pages 432 – 439. North Holland, 1987.

[71] R. Parikh. Logical omniscience. In D. Leivant, editor, *Springer Lecture Notes in Computer Science no. 960*, pages 22–29. Springer-Verlag, 1995.

[72] Rohit Parikh. Propositioanl logics of programs: new directions. In M. Karpinski, editor, *Foundations of Computation Theory*, number 158 in LNCS, pages 347 – 359. Springer, 1983.

[73] Rohit Parikh. The logic of games and its applications. In M. Karpinski and J. van Leeuwen, editors, *Topics in the Theory of Computation*, volume 24 of *Annals of Discrete Mathematics*. Elsevier, 1985.

[74] Rohit Parikh. Language as social software (abstract). In *International Congress on Logic, Methodology and Philosophy of Science*, page 415, 1995.

[75] Rohit Parikh. Lanuage as social software. In S. Shieh J. Floyd, editor, *Future Pasts: The Analytic Tradition in Twentieth Century Philosophy*, pages 339–350, 2001.

[76] Rohit Parikh. Social software. *Synthese*, 132:187–211, September 2002.

[77] Rohit Parikh. Towards a theory of social software. In *Proceedings of DEON 2002*, pages 187–211, September 2002.

[78] Rohit Parikh. Levels of knowledge, games, and group action. *Research in Economics*, 57(3):267 – 281, 2003.

[79] Rohit Parikh and Paul Krasucki. Levels of knowledge in distributed computing. *Sadhana - Proc. Ind. Acad. Sci.*, 17:167 – 191, 1992.

[80] Rohit Parikh and Eric Pacuit. Safe votes, sincere votes, and strategizing. Working Paper.

[81] Rohit Parikh, Eric Pacuit, and Eva Cogan. The logic of knowledge based obligation. *Knowledge, Rationality and Action*, forthcoming.

[82] Rohit Parikh and R. Ramanujam. Distributed processes and the logic of knowledge. In *Logic of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 256 – 268. Springer, 1985.

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

[84] Marc Pauly. Programming and verifying subgame perfect mechanisms. to appear in the *Journal of Logic and Computation.*

[85] Marc Pauly. *Logic for Social Software.* Illc dissertation series 2001-10, University of Amsterdam, 2001.

[86] Marc Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149 - 166, 2002.

[87] Marc Pauly. On the complexity of coalitional reasoning. *International Game Theory Review*, 4(3):237 - 254, 2002.

[88] Marc Pauly and Mike Wooldridge. Logic for mechanism design - a manifesto. Unpublished manuscript.

[89] Jan Plaza. Logics of public communications. In *Proceedings, 4th International Symposium on Methodologies for Intelligent Systems*, 1989.

[90] V. R. Pratt. Semantical considerations on floyd-hoare logic. In *Proc. 17th Symp. Found. Comput. Sci.*, IEEE, pages 109 - 121, 1976.

[91] W. Riker and P. Ordeshook. A theory of the calculus of voting. *American Political Science Review*, 62:25 - 42, 1968.

[92] Ariel Rubinstein. *Economics and Language.* Cambridge University Press, Cambridge, 2000.

[93] Samer Salame. *Majority Logic.* PhD thesis, CUNY Graduate Center, 2005.

[94] Tuomas Sandholm. Making markets and democracy work: A story of incentives and computing. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 1649–1671, 2003.

[95] Mark Satterthwaite. *The Existence of a Strategy Proof Voting Procedure: a Topic in Social Choice Theory.* PhD thesis, University of Wisconson, 1973.

[96] Mark Satterthwaite. Strategy-proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[97] Y. Shoham and M. Tennenholtz. Non-cooperative computing: Boolean functions with correctness and exclusivity. *Journal of Theoretical Computer Science*, to appear.

[98] R. Stalnaker. The problem of logical omniscience, I. *Synthese*, 89:425 – 440, 1991.

[99] R. Stalnaker. *Context and Content*, chapter The Problem of Logical Omniscience, II. Oxford: Clarendon Press, Oxford, 1999.

[100] J. van Benthem. Games in dynamic epistemic logic. *Bulletin of Economic Research*, 53:216 – 248, 2001.

[101] Johan van Benthem. 'one is a lonely number': on the logic of communication. Available at http://staff.science.uva.nl/~johan/Muenster.pdf.

[102] W. van der Hoek and M. Wooldridge. Towards a logic of rational agency. *L. J. of the IGPL*, 11(2):135 – 159, 2003.

[103] R. van der Meyden. Axioms for knowledge and time in distributed systems with perfect recall. In *Proc. IEEE Symposium on Logic in Computer Science*, pages 448–457, July 1994.

[104] R. van der Meyden and K. Wong. Complete axiomatizations for reasoning about knowledge and branching time. *Studia Logica*, 75(1):93 – 123, October 2003.

[105] Ron van der Meyden. The dynamic logic of permission. *Journal of Logic and Computation*, 6(3):465–479, 1996.

[106] Hans van Ditmarsch. *Knowledge Games.* PhD thesis, University of Gronigen, 2000.

[107] Yde Venema. Temporal logic. In Lou Goble, editor, *The Blackwell Guide to Philosophical Logic.* Blackwell Philosophy Guides, 2001.

[108] Angela Weiss and Rohit Parikh. Completeness of certain bimodal logics of subset spaces. *Studia Logica*, 71(1):1–30, 2002.

[109] Michael Wooldridge. *Reasoning about Rational Agents.* The MIT Press, 2000.

[110] Anna Maria Zanaboni. Reasoning about knowledge: Notes of rohit parikh's lectures. Published in Italy: Cassa di Risparmio di Padova e Rovigo, June 1991.

Based on lectures given at the 3rd International School for Computer Science Researchers, Acireale, June 1991.