Some Comments on History Based Structures

Eric Pacuit

ILLC, University of Amsterdam
epacuit@cs.gc.cuny.edu

Abstract. History based models, introduced by Parikh and Ramanujam, provide a natural mathematical model of social interactive situations. These models offer a "low level" description of a social situation — describing the situation in terms of events, sequences of events, and the agents' view of these events. A multi-agent epistemic temporal modal logic can be used to reason about these structures. A number of other models have been proposed in the literature which can be used as a semantics for such a logical language. Most notably, the interpreted systems discussed by Fagin et al. In this paper, we will discuss the differences and similarities between these two mathematical models. In particular, it is shown that these two semantics are modally equivalent. We will conclude with a discussion of a number of questions that are raised when history based models are used to reason about game-theoretic situations.

1 Introduction

History based models, introduced by Parikh and Ramanujam [1, 2], provide a natural model of social interactive situations relevant for the analysis of *social software*. The key idea behind social software is that 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 [3], and has recently gained the attention of a wide range of research communities, including computer scientists, game theorists and philosophers. Starting with [3] and more recently in [4–6], 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 paper, the fact that a group of agents is somehow involved in the execution of a social procedure will play an essential role.

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. In this paper, we will focus only of the first category — mathematical models of social situations. The recent work of Marc Pauly [7] discusses issues relevant to the second category (a theory of correctness of social procedures). There has been a wealth of literature devoted to designing social procedures. In particular, fair division algorithms and voting procedures (see [8] for more information).

The objective of this paper is to discuss a mathematical model of multi-agent social interactive situations appropriate for the analysis of social software. 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.

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 Hintikka's seminal book, *Knowledge and Beliefs* [9], 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 ([10, 11]), economists ([12–14]) and philosophers ([15]). 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 [16] and [10] 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 paper 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 3 shows how this basic framework can be extended to provide a model for multi-agent epistemic temporal logics. Section 4 contains the main technical result of thsi paper. Finally, we conclude with an extended discussion of future work.

2 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. The framework described in this chapter is based on that of Parikh and Ramanajam [1, 2].

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 [1, 2], 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, CTL^*, ATL and ATL^* can be used to reason about situations where there may not be a unique next event. The reader is referred to [17, 18] for information on temporal logic and [19] 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.

3 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 paper 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.

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:

[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 [20]. For example, the string abc is embeddable in $aabbaaca$, but abc is not embeddable in $bbaac$.

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.*

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 [22], 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 $He \in \mathcal{H}$,

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

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

---

[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: $K_i(\phi \to \psi) \to (K_i\phi \to K_i\psi)$, $\mathcal{K}_i\phi \to \phi$ 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: $K_i(\phi \to \psi) \to (K_i\phi \to K_i\psi)$, $\mathcal{K}_i\bot \to \bot$, $K_i\phi \to K_iK_i\phi$ and $\neg K_i\phi \to K_i\neg K_i\phi$. More information can be found in [21].

[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 [22] for more information.

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 [22], using a slightly different framework. The precise connection between the two frameworks will be discussed below. We first report the relevant results from [22]. 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 [10] for a proof).

PC. All tautologies of propositional logic
K2. $K_i(\phi \to \psi) \to (K_i\phi \to K_i\psi)$
K3. $K_i\phi \to \phi$
K4. $K_i\phi \to K_iK_i\phi$
K5. $\neg K_i\phi \to K_i\neg K_i\phi$
MP. From ϕ and $\phi \to \psi$ infer ψ
N. From ϕ infer $K_i\phi$

Call this axiom system $\mathbf{S5}_n$. The following axiom system is from [22] is used to reason about (linear) time.

T1. $\bigcirc\phi \wedge \bigcirc(\phi \to \psi) \to \bigcirc\psi$
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 $\phi' \to \neg\psi \wedge \bigcirc\phi'$ infer $\phi' \to \neg(\phi\ U\psi)$

A few remarks about the rule $RT2$. This rule is equivalent to the following simpler two rules:

RT2$_1$. From $\phi_1 \to \psi_1$ and $\phi_2 \to \psi_2$ infer $(\phi_1 U \psi_2) \to (\phi_2 U \psi_2)$
RT2$_2$. From $\bigcirc\phi \to \phi$ infer $F\phi \to \phi$

To see that RT2 follows from these rules, suppose that we have derived $\phi' \to \neg\psi$ and $\phi' \to \bigcirc\phi'$. Then, using standard propositional reasoning and T2 we can infer $\bigcirc\neg\phi' \to \neg\phi'$. Hence using RT2$_2$ we can infer $F\neg\phi' \to \neg\phi'$, i.e., $(\top U \neg\phi') \to \neg\phi'$. Now notice that for any formula ϕ, we can derive $\phi \to \top$ and using propositional reasoning we can infer $\psi \to \neg\phi'$. Thus using RT2$_1$, we can infer $(\phi U \psi) \to (\top U \neg\phi')$; and so using propositional reasoning we can conclude $\neg(\phi U \psi)$. Showing RT2$_1$ and 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 [22] that will be of interest are:

KT1. $K_i \bigcirc \phi \to \bigcirc K_i\phi$
KT2. $K_i\phi_1 \wedge \bigcirc(K_i\phi_2 \wedge \neg K_i\phi_3) \to L_i((K_i\phi_1)\ U[(K_i\phi_2)\ U\neg\phi_3])$

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 $K_iG\phi \to GK_i\phi$ – 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 [23]. In [22], 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 [22]) refer to [22].

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 [24] 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 [24] 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 [25]. 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 [26], van der Meyden and Wong prove a series of completeness results for logics of knowledge with branching time operators.

4 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. [24, 10, 22] interpreted systems.

We begin by formally defining interpreted systems. The reader is referred to [10] and [22] 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 $r : \mathbb{N} \to L^{n+1}$ $r(t)$ 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 $\pi : (\mathcal{R} \times \mathbb{N}) \times \mathsf{At} \to \{\mathbf{true}, \mathbf{false}\}$, 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 [22] 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.

Lemma 1. *For each infinite history $H, H' \in \mathcal{H}$, for each $t, m \in \mathbb{N}$, $H_t \sim_i H'_m$ iff $(r_H, t) \sim_i (r_{H'}, m)$.*

Proof. Straightforward. □

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

Theorem 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 [22] 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 [22] 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.

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 $\mathcal{H}_\mathcal{R}$

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

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.

Lemma 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 $H^r_t \sim_i H^{r'}_m$.*

Proof. Straightforward. ◻

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 $\pi(r,t)(p) = \textbf{true}$. The proof of the following theorem which shows that our translation works as intended is similar to the above theorem and so is ommitted.

Theorem 2. *Let $(\mathcal{R}, \pi)$ be an interpreted system. Then for each $r \in \mathcal{R}$ and each $\phi \in \mathcal{L}^{KT}_n$*

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

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 [22] 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 [10] 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 [2], 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.

5 Conclusion and Future Work

This paper discusses a formal framework intended to be used to represent social interactive situations relevant for the analysis of social procedures. 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. 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. In general, some of the events may be caused by an agent, i.e., an agent can perform a particular action, and others may be caused by nature (which can be viewed as a special type of agent). See [27] for more information about using this framework for studying social software.

We have shown that the history based models of Parikh and Ramanujam are modally equivalent (with respect to the language $\mathcal{L}_n^{KT}$) to the interpreted systems of Fagin et al. The proof is technically straightforward; however it does contain some insights. Namely, it shows a real difference from the modeler's point of view between these two semantics. The history based structures of Parikh and Ramanujam are appropriate for formalizing many of the issues relevant for the analysis of social software. We now move on to a discussion of a future research direction.

In the game theory literature, *extensive game forms* are used to model the decision problems encountered by agents in strategic situations [28]. These structures have much in common with history based structures. In fact, as will be shown below, 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 3. Reasoning about extensive games from this is point of view is explored in a recent paper of Bonanno [29] (cf. [30]). 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 3 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 finite global history H, $\mathsf{last}(H)$ denotes the last 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* $H' \in \mathcal{F}(H_t)$*,* $\mathsf{last}(H'_{t+1}) \in A_i$*.*

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. For this point of view, it is very natural to use these structures to interpret a suitable modal language. Indeed there is a long list of researchers in both game theory and logic who have noticed this fact (see [31–33, 29, 30, 34] for recent contributions). The focus of future research will be to extend a history based game form with local view functions for each agent. We can then extend the results of [31] by bringing in epistemic modal operators for each agent.

References

1. Parikh, R., Ramanujam, R.: Distributed processes and the logic of knowledge. In: Logic of Programs. Volume 193 of Lecture Notes in Computer Science., Springer (1985) 256 – 268
2. Parikh, R., Ramanujam, R.: A knowledge based semantics of messages. Journal of Logic, Language and Information **12** (2003) 453 – 467
3. Parikh, R.: Language as social software (abstract). In: International Congress on Logic, Methodology and Philosophy of Science. (1995) 415
4. Parikh, R.: Social software. Synthese **132** (2002) 187–211
5. Parikh, R.: Towards a theory of social software. In: Proceedings of DEON 2002. (2002) 187–211
6. Parikh, R.: Lanuage as social software. In J. Floyd, S.S., ed.: Future Pasts: The Analytic Tradition in Twentieth Century Philosophy. (2001) 339–350
7. Pauly, M.: Programming and verifying subgame perfect mechanisms. to appear in the *Journal of Logic and Computation* (to appear)
8. Brams, S.J., Taylor, A.D.: Fair Division: From cake-cutting to dispute resolution. Cambridge University Press, Cambridge (1996)
9. Hintikka, J.: Knowledge and Belief: An Introduction to the Logic of the Two Notions. Cornell University Press, Ithaca (1962)

10. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. The MIT Press, Boston (1995)
11. Zanaboni, A.M.: Reasoning about knowledge: Notes of rohit parikh's lectures. Published in Italy: Cassa di Risparmio di Padova e Rovigo (1991) Based on lectures given at the 3rd International School for Computer Science Researchers, Acireale, June 1991.
12. Bonanno, G., Battigalli, P.: Recent results on belief, knowledge and the epistemic foundations of game theory. Research in Economics **53** (1999) 149–225
13. Bonanno, G., Nehring, K.: Agreeing to disagree: a survey. Document prepared of an invited lecture at the Workshop on Bounded Rationality and Economic Modelling (1997)
14. Aumann, R.: Interactive epistemology I: Knowledge. International Journal of Game Theory **28** (1999) 263–300
15. Gochet, P., Gribomont, P.: Epistemic logic. Forthcoming in *The Handbook of History and Philosophy of Logic*, Elsevier, edited by D. Gabbay and J. Woods (to appear)
16. Parikh, R.: Logical omniscience. In Leivant, D., ed.: Springer Lecture Notes in Computer Science no. 960. Springer-Verlag (1995) 22–29
17. Venema, Y.: Temporal logic. In Goble, L., ed.: The Blackwell Guide to Philosophical Logic. Blackwell Philosophy Guides (2001)
18. Goranko, V.: 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) (to appear)
19. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT Press, Boston (1999)
20. Parikh, R., Krasucki, P.: Levels of knowledge in distributed computing. Sadhana - Proc. Ind. Acad. Sci. **17** (1992) 167 – 191
21. Chellas, B.: Modal Logic: An Introduction. Cambridge University Press, Cambridge (1980)
22. Halpern, J., van der Meyden, R., Vardi, M.: Complete axiomatizations for reasoning about knowledge and time. SIAM Journal of Computing **33** (2004) 674 – 703
23. van der Meyden, R.: Axioms for knowledge and time in distributed systems with perfect recall. In: Proc. IEEE Symposium on Logic in Computer Science. (1994) 448–457
24. Halpern, J., Vardi, M.: The complexity of reasoning about knowledge and time. J. Computer and System Sciences **38** (1989) 195 – 237
25. French, T., van der Meyden, R., Reynolds, M.: 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 (to appear)
26. van der Meyden, R., Wong, K.: Complete axiomatizations for reasoning about knowledge and branching time. Studia Logica **75** (2003) 93 – 123
27. Pacuit, E.: Topics in Social Software: Information in Strategic Situations. PhD thesis, City University of New York Graduate Center (2005)
28. Osborne, M., Rubinstein, A.: A Course in Game Theory. The MIT Press, Boston (1994)
29. Bonanno, G.: A characterization of von neumann games in terms of memory. Synthese **139** (2004) 237–256
30. van Benthem, J.: Games in dynamic epistemic logic. Bulletin of Economic Research **53** (2001) 216 – 248
31. Harrenstein, P., van der Hoek, W., Meyer, J.J., Witteveen, C.: A modal characterization of nash equilibrium. Fundamenta Informaticae **57** (2003) 281 – 321
32. van Otterloo, S.: Reasoning about extensive games. Available at (2005)
33. van Otterloo, S., van der Hoek, W., Wooldridge, M.: Preferences in game logics. In: Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), New York (2004)
34. van Benthem, J.: Extensive games as process models. (2001)