2003

TR-2003002: A Knowledge Based Semantics of Messages

Rohit Parikh

R. Ramanujam

A Knowledge based semantics of messages

Rohit Parikh

Department of Computer Science,*
CUNY Graduate center
365 Fifth Avenue
New York, NY 10016-4309, USA.
E-mail: ripbc@cunyvm.cuny.edu

R. Ramanujam

The Institute of Mathematical Sciences
C.I.T. Campus, Chennai 600 113
India.
E-mail: jam@imsc.res.in

If a lion could talk, we could not understand him.
Ludwig Wittgenstein

Abstract

We investigate the semantics of messages, and argue that the meaning of a message is naturally and usefully given in terms of how it affects the *knowledge* of the agents involved in the communication. We see that the semantics depends on the protocol used by the agents, and this leads us to knowledge based specification of protocols. While these notions are natural for distributed computations, we suggest that the considerations discussed here may be relevant in more general linguistic contexts.

1 Introduction

In natural as well as formal languages, messages have an interesting semantic status. Syntactically, they are perhaps no different from utterances, perhaps by speakers

*Also, Department of Computer Science, Brooklyn College. Research supported by NSF and a grant from the PSC-CUNY FRAP program

not present when the utterance is heard or received. But when we designate any communication as a 'message', we invest it with a purpose, an accomplishment of specific semantic objectives. Deriving the meaning of a message may well involve much more than the syntactic structure of the message.

The famous lines from Longfellow's poem "Paul Revere's ride", run:

> *He said to his friend, "if the British march*
> *by land or sea from the town tonight,*
> *Hang a lantern aloft in the belfry arch*
> *Of the North church tower as a signal light, –*
> *One, if by land, and two, if by sea;*
> *And I on the opposite shore will be, ...*

It is obvious that Paul Revere is setting up a protocol with his friend whereby a signal with two possible values can be used to indicate one of two alternatives. This example illustrates several issues any semantics of messages needs to address. While the hanging of a lantern is to show a light, doing this at a specific time, in a specific 'state of the world', by a specific person, carries a unique meaning to one who sees the light, particularly when the two have agreed on a protocol for signalling, and the latter trusts the former to follow the protocol.

Indeed, there is more. The meaning may also depend on the medium of communication in which the message passing takes place. If Paul Revere and his friend had accurate, synchronized watches, much more information, e.g. the size of the British force, could have been conveyed. To see this, suppose I want a friend to send me information that will distinguish between a thousand possibilities, but my phone (or his) is tapped by enemies, and he dare not talk to me. Here is our protocol. He calls me, lets the phone ring once or twice, and hangs up. I then count the number of minutes elapsed since 8 AM (he calls between 8 AM and 5 PM), which gives me 540 possibilities. The signal has two values, so in all that makes 1080, ample for my purpose. Thus even though the signal has only two values, in the context it conveys a little over 10 bits of information.

We spoke of the receiver of the message interpreting it according to an agreed protocol. In fact, the message itself may be part of setting up the protocol between the parties in a bootstrapping manner. Moreover, trust may not be necessary to give meaning; the message may itself be for establishing trust in the receiver about the sender. Consider the following case. A secret group which trusts only its own members, is holding a meeting. A member who wants to attend the meeting must identify himself by saying "rain in Spain is mainly in the plain". Now clearly this message does not have the usual meaning of that sentence. We cannot talk of the knowledge the sentence creates in the presence of trust, since the purpose of this message is to create trust. For the same reason, a letter of recommendation written by an applicant himself does not create the same knowledge in the prospective employer as an identical letter written by an unbiased authority.

Essentially, all these considerations suggest that evaluating the meaning of a message in a manner that is dependent on the protocol used by the communicating parties gives us implicature as well. Then there is the additional dimension of dealing with uncertainty. Of course, when the communicating parties are human beings, utilities are also needed to evaluate meaning and implicature. Even in the case of machines which may not have utilities, the designers of those machines do, and strategic considerations may well bring utilities into the semantics of messages.

In this paper, we focus on distributed systems of computing agents that communicate (only) by message passing. We study how messages may be interpreted in such systems, and the study illustrates how the informational value of the message depends also on the protocol, without use of utilities. We suggest that a knowledge based semantics of messages not only offers a solution to the difficulties listed above, but also suggests a semantic specification of protocols using knowledge assertions.

The last point, about semantic specification of protocols, is important, and highlights the difference between *extensional* and *intensional* treatment of protocols, a difference that is critical in the context of utilities. We have been discussing how the meaning of a message is dependent on the protocol used, but how is the protocol itself presented ? If a protocol is seen (extensionally) as a set of possible evolutions of the considered system, such a set may be used in the semantics of messages. However, we need a mechanism that generates these evolutions, which requires an intensional description. In the context of *distributed protocols* studied in computer science, a protocol is simply a bunch of rules describing what messages each agent may send under different circumstances, as well as the response of agents to the receipt of messages (see [T94]). This may seem circular: the protocol is described using messages, and the messages are interpreted according to the protocol. Knowledge based specification of protocols offers a way out of such circularity, as we will see, see also [P95].

We suggest that knowledge based semantics of messages is also useful for the theory of distributed systems. These are systems typically designed to achieve specific algorithmic goals, and the concern is about efficient management of resources (number of rounds of communications, length of messages etc). A general semantic theory, which studies the transition structure of messages, is lacking. On the other hand, concurrency theory does study the interaction of system structure and behaviour in an abstract setting and thus provides semantic models, but offers no theories of communication comparable in richness to that of theories of causal independence.

In the context of a distributed system, it is easy to see the role of a message. An agent in a distributed system, at any local state, has only a *partial* view of the system state. The purpose of a message is to offer the receiver an enhanced view of the system state. Viewed thus, we can at once see a number of issues to be addressed by any semantic theory of communication:

- It is easily seen that receipt of a message often causes a change of system view. Does sending of a message also change an agent's view ? When agents communicate by the hand-shaking mode of synchronization, sending a message

can certainly cause a view update. Even in the context of asynchronous message passing, the sender can reason that on receipt of the message, the receiver's view would be updated at that state and record this information. Such reasoning can be found in knowledge-based analyses of distributed protocols [FHMV95].

- The same message, sent by different agents, or received by different agents, may cause different changes in system behaviour. Thus the semantics of messages may depend on how agents are defined, and their behaviour in turn depends on what messages mean to them.
- We can easily conceive of situations where an agent receiving the same message at different states acts differently. Thus the meaning of a message is also dependent on the receiver's state.
- Clearly, messages are interpreted according to the protocol followed (implicitly) by the agents, and systems where agents agree to change the protocol dynamically do exist !
- Distributed protocols typically contain not only assumptions about agents sending messages according to the protocols, but also actions to be taken when agents' messages violate the protocols. This is particularly problematic for semantic models.
- Message based systems use a menagerie of message categories – signals, interrupts, resource requests, data messages, control messages, and so on, and implementations never treat them the same way. On the other hand, a semantic model should perhaps not even consider messages as concrete objects, but yet offer possibilities of modelling such behavioural variations.
- Suppose a single message m has the same effect as a sequence of messages σ. While m and σ must needs be distinguished from an algorithmic point of view, they should both perhaps be identified as having the same denotation.
- What is communicated by the message may be not only current information, but also *temporal* information about possible future behaviour.

Our basic technical tool is as follows:

- States are specified by properties stated in a logical language and the view of an agent at a state can be thought of as those global properties which are *known* to that agent at that state.
- The denotation of a communication is then a set of pairs of the form (α, β) such that if, at any local state of an agent s, the global property α is known by the agent, then in the next state of the agent, the global property β will be known by that agent.

In what follows, we present a logical framework in which this idea is developed formally.

2 An abstract model

We will first consider an abstract extensional presentation of a distributed system, in which the system is described as a set of *global histories*, each of which represents one possible system evolution given by a sequence of global events. For each system, the set of *agents* that participate in its computations is assumed to be a fixed finite set. Similarly, for each system, the set of possible global events is fixed.

For convenience, we fix $n > 0$, and consider only systems with agents from $[n] = \{1, 2, \ldots, n\}$, and events come from a fixed (possibly infinite) set E. E^* is the set of all finite sequences over E and E^ω is the set of all infinite sequences over E; we will let $h, h', \ldots$ range over the former, and $H, H', \ldots$ over the set $E^* \cup E^\omega$. Let $H \preceq H'$ denote that H is a finite prefix of H'. We write hH' to denote the concatenation of finite history h with the possibly infinite history H'. When H is infinite or of length $\geq k$, we let H_k denote the finite prefix of H consisting of the first k elements. For a set $\mathcal{H}$, let $\mathcal{P}(\mathcal{H})$ denote the set $\{h \mid h \preceq H \text{ for some } H \in \mathcal{H}\}$ containing all finite prefixes of sequences in $\mathcal{H}$.

Definition 2.1 *A* **system** *is a tuple $S = (\mathcal{H}, \lceil_1, \ldots, \lceil_n)$, where $\mathcal{H} \subseteq E^\omega$ is the set of all (infinite) possible global histories of S, and for $i \in [n]$, $\lceil_i \colon \mathcal{P}(\mathcal{H}) \to E^*$ is the* projection map *for i. $\mathcal{H}_i \stackrel{\text{def}}{=} \{\lceil_i(h) \mid h \in \mathcal{P}(\mathcal{H})\}$ is the set of local histories of i.*

Thus local histories are got by projecting global histories to local components. Note that the definition is very general, and we may want more conditions to apply usually: for instance, if $h_1 \preceq h_2 \preceq H \in \mathcal{H}\}$, then we would expect that $\lceil_i(h_1) \preceq \lceil_i(h_2)$ as well, but this is not enforced by the definition above. The points we wish to illustrate may already be highlighted with such a general description.

Definition 2.2 *Let h, h' be finite global histories in $\mathcal{H}$. For $i \in [n]$, define $h \sim_i h'$ iff $\lceil_i(h) = \lceil_i(h')$.*

Clearly, $\sim_i$ is an equivalence relation, and it gives the *indistinguishability relation for i*. We can consider this relation as giving the information partition for i in the system S; that is, given the information available to i, the histories h and h' cannot be disntinguished. Note again that we may have histories h_1, h_2, h such that $h_1 \not\sim_i h_2$ but $h_1 h \sim_i h_2 h$ (constituting a kind of 'forgetting').

The properties of such systems can be studied in a logical language. Let L be a language which has formulae expressing (time dependent) properties of global histories. Then we can write $H, k \models A$, for A belonging to L, to mean that the history H satisfies formula A at time k. If the truth value of A does not depend on k, then it is timeless. If A has the property that once true it remains true, then it is persistent. We expand L to a larger language LK by closing under boolean connectives and operators K_i. Thus if A is a formula of LK and i is a process, then $K_i(A)$, meaning i knows A, is also in LK. We can then define $H, k \models K_i(A)$ to hold if for all m and

all $H' \in \mathcal{H}$, if $H'_m \sim_i H_k$ then $H', m \models A$. Clearly what the process i knows at time k depends only on its local history. Moreover, the laws of logic $LK5$ (the $S5$ version of the logic of knowledge) are valid.

For definiteness, we fix a specific language L so that the semantics of $H, k \models A$ is also fixed. Since the basic elements of the model are sequences, a linear time temporal logic suggests itself. Let $P = \{p_0, p_1, \ldots\}$ be a countable set of atomic propositions. Formally, the syntax of the logic is given by:

$$\alpha, \beta \in \mathcal{L}_0 ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta \mid \bigcirc\alpha \mid \alpha \mathbf{U} \beta \mid \mathbf{K}_i\alpha$$

A **model** is a pair $M = (S, V)$, where $V : \mathcal{P}(\mathcal{H}) \to 2^P$ is a valuation map on finite prefixes of global histories which gives the truth values of some atomic predicates at the states. We can now inductively define the notion $H, k \models \alpha$, for $H \in \mathcal{H}$, $k \geq 0$ and $\alpha \in \mathcal{L}_0$:

- $H, k \models p$ iff $p \in V(H_k)$, for $p \in P$.
- $H, k \models \neg\alpha$ iff $H, k \not\models \alpha$.
- $H, k \models \alpha \vee \beta$ iff $H, k \models \alpha$ or $H, k \models \beta$.
- $H, k \models \bigcirc\alpha$ iff $H, k+1 \models \alpha$.
- $H, k \models \alpha \mathbf{U} \beta$ iff there exists $m \geq k$ such that $H, m \models \beta$ and for all $\ell : k \leq \ell < m$, $H, \ell \models \alpha$.
- $H, k \models \mathbf{K}_i\alpha$ iff for all $m \geq 0$, for all $H' \in \mathcal{H}$ such that $H_k \sim_i H'_m$, $H', m \models \alpha$.

The formula α is said to be *satisfiable* if there exists a model M, a global history $H \in \mathcal{H}$ in M and $k \geq 0$ such that $M, k \models \alpha$. α is said to be *valid* iff $\neg\alpha$ is not satisfiable. The following formulas, the laws of logic $LK5$, are easily seen to be valid:

- $\mathbf{K}_i(\alpha \supset \beta) \supset (\mathbf{K}_i\alpha \supset \mathbf{K}_i\beta)$.
- $\mathbf{K}_i\alpha \supset \alpha$.
- $\mathbf{K}_i\alpha \supset \mathbf{K}_i\mathbf{K}_i\alpha$.
- $\neg\mathbf{K}_i\alpha \supset \mathbf{K}_i\neg\mathbf{K}_i\alpha$.

We are now ready to present the knowledge based semantics of messages in a system S. For our purpose, a message is simply an element $m \in E$.

Definition 2.3 $Sem_S(i, m) \stackrel{\text{def}}{=} \{(\alpha, \beta) \mid (\forall H \in \mathcal{H})(\forall k \geq 0)(H, k \models \mathbf{K}_i\alpha \rightarrow (\forall h' \in \mathcal{P}(\mathcal{H})(H_k \preceq h', |h'| = \ell \geq k \wedge \lceil_i(h') = \lceil_i(H_k)e)(H'\ell \models \mathbf{K}_i\beta\}$,
i.e. $\{(\alpha, \beta) \mid$ *for all* $H \in \mathcal{H}$, *for all* $k \geq 0$, *if* $H, k \models \mathbf{K}_i\alpha$ *then, for all* $h' \in \mathcal{P}(\mathcal{H})$ *such that* $H_k \preceq h'$, h' *is of length* $\ell \geq k$ *and* $\lceil_i(h') = \lceil_i(H_k)e$, *we have* $H', \ell \models \mathbf{K}_i\beta\}$.

Intuitively, if i knows α before the message m then it always knows β after it. Thus $Sem(i,m)$ is a subset of $\mathcal{L}_0 \times \mathcal{L}_0$, and every pair (α,β) in it can be seen as a *view transformer* for agent i: if knowledge of α is part of the view that i has of the global system state before the communication event, then knowledge of β is part of its view after the communication.

It is easy to see that the definition can be generalized to $Sem_S(i,h)$, where $h \in E^*$: $Sem_S(i,h) \stackrel{\text{def}}{=} \{(\alpha,\beta) \mid$ for all $H \in \mathcal{H}$, for all $k \geq 0$, if $H,k \models \mathbf{K}_i\alpha$ then, for all $h' \in \mathcal{P}(\mathcal{H})$ such that $H_k \preceq h'$, h' is of length $\ell \geq k$ and $\lceil_i(h') = \lceil_i(H_k)h$, we have $H',\ell \models \mathbf{K}_i\beta\}$. The following proposition records some simple observations that follow from the validities listed earlier.

Proposition 2.4 *1. If $(\alpha,\beta) \in Sem_S(i,h)$ and $\gamma_1 \supset \alpha$, $\beta \supset \gamma_2$ are valid formulas, then $(\gamma_1,\gamma_2) \in Sem_S(i,h)$ as well.*

2. If $(\mathbf{K}_i\alpha,\beta) \in Sem_S(i,h)$, then $(\alpha,\mathbf{K}_i\beta) \in Sem_S(i,h)$ as well.

3. If $(\mathbf{K}_i\alpha,\beta_1) \in Sem_S(i,h)$ and $(\mathbf{K}_i\alpha,\beta_2) \in Sem_S(i,h)$, then $(\alpha,\mathbf{K}_i(\beta_1 \wedge \beta_2)) \in Sem_S(i,h)$ as well.

4. If $(\alpha,\beta) \in Sem_S(i,h)$ and $(\beta,\gamma) \in Sem_S(i,h')$ then $(\alpha,\gamma) \in Sem_S(i,hh')$.

Condition 1) above is *monotonicity* and holds because we regard all possible global histories as equal. However, if the possible global histories are ordered by plausibility, then monotonicity will fail. I.e. the most plausible interpretations of the message m in the presence of α (and in which β does hold) might no longer be possible in the presence of the stronger formula γ and in that case we can no longer rely on β holding.

Thus we can use knowledge formulas to describe changes effected by a sequence of events; dually, we may consider such pairs of formulas as a specification, and ask what sequence of events implements the specification.

Proposition 2.5 *Suppose we are given two systems $S = (\mathcal{H},\lceil_1,\ldots,\lceil_n)$ and $S' = (\mathcal{H}',\lceil_1,\ldots,\lceil_n)$ which are identical in their structure except that $\mathcal{H}' \subseteq \mathcal{H}$. Let α,β be two formulas in which all occurrences of the operators $\mathbf{K}_i$ are positive, i.e. occur under no negation signs or under an even number of them. Then if (α,β) is in $Sem_S(i,m)$, it is also in $Sem_{S'}(i.m)$.*

The proof is straightforward using the fact that the quantification over histories of S' is always over a smaller set and hence more likely to be true. Thus given a set of conventions which restrict the set os possible histories, more information can be acquired through a message. This of course need not hold with negative knowledge. It may happen that in a larger $\mathcal{H}$, after message m is received, some $\neg\mathbf{K}_i(\beta)$ holds, but that with a smaller $\mathcal{H}'$, $\mathbf{K}_i(\beta)$ does hold and hence $\neg\mathbf{K}_i(\beta)$ fails.

For a very simple example, if you tell me β but the global histories in $\mathcal{H}$ allow the possibility of your lying, then I do not know β. If we restrict the global histories to those in which you never lie, then I do know β.

We can observe that our semantics satisfies many of the criteria set out in the last section. The denotation of a message is dependent on the protocol in a crucial manner. For instance, fix a system S and let m be an event such that for all $h, h' \in \mathcal{P}(\mathcal{H})$, if $\lceil_i(h') = \lceil_i(h)m$, then $h' = hm'm$; that is, though m' is external to i, or 'invisible' to i, when m occurs, i gets the information that m' has occurred as well.

Note that the only condition that the Sem definition insists on is that the event m be 'visible' to i via the $\lceil_i$ map. Thus, when m is the sending of a message in an asynchronous system, our semantics allows for the event to have non-trivial meaning for the sender as well. This is important in protocols where the sending by itself starts a chain of events in all histories, without the sender receiving any later information explicitly. For example what you knew before sending a nasty email to your boss, e.g., that he regarded your job as secure, might no longer be true after you send that message, even though you have not, so far, heard anything from him.

Moreover, it can be seen that in this framework, the meaning of a message may vary depending on the identity of the sender or the receiver, their state (at the time of sending or receiving), and on the medium of communication. The event m in $Sem(i, m)$ may also be a synchronous 'hand-shake' type of communication involving more than one agent, and the meaning may be different for each of the participating agents. Finally, code messages like "rain in Spain is mainly in the plain" that seek to establish trust among agents may also be given meaning in a uniform manner.

Thus we see that even though the messages are just strings, and do not necessarily belong to a language for which we **already** have a semantics at hand, we can still assign a semantics to them. The situation is a little like the one where we arrive on some strange planet and do not know the language nor is there any interpreter. After some experience we learn to interpret your language. Following Quine, we might never know for sure whether your word *gavagai* refers to a rabbit or a rabbit part. But if we do not, it is because the histories in which rabbit parts appear and those in which rabbits appear are the same and hence *as properties of histories* these are indentical.

2.1 Formulas as messages

It is interesting to consider systems where the messages sent are themselves formulas of the logic considered. This requires a notion of *honesty* among agents. Fix a system S. As in [Pa], call an agent i honest iff for all $h \in \mathcal{P}(\mathcal{H})$, if $h(send_i\ \alpha) \in \mathcal{P}(\mathcal{H})$ then $h \models \mathbf{K}_i\alpha$. Thus, when an agent j receives the message, she knows that α was true when it was sent, and if α is a persistent formula, then j knows that it is still true. (A formula α is said to be persistent, if, whenever $h \models \alpha$, then for all h' such that $h \preceq h'$, we have: $h' \models \alpha$.)

[HF] define a processor to be honest if it sends a formula only when it knows it to be true. If one receives the formula γ from such an honest processor as message then we would expect that $(\alpha, \beta) \in Sem(i, \gamma)$ iff β is a logical consequence of $\alpha \wedge \gamma$. If $\alpha \wedge \gamma$ is inconsistent or if γ is not persistent then complexities arise.

Moreover, there are other possible definitions of honesty. One is that a message should be known to be true when received rather than when sent. For example, if a boy knows that his girl friend, at another college, always reads his letters under her favourite tree, he may start his letter with "You are now sitting under your favourite tree, reading this letter ... ", and we may not want to say that he is dishonest just because that statement was not true at the time he wrote that letter. In fact, if the postal system is unreliable, then the message is true only *if* and when it is received.

To give an example (due to Dexter Kozen), of a non-persistent formula if I tell you the formula γ where γ is: "you don't know this, but there is a bug crawling on your shirt". Then in this case γ as a message does have the semantic value of a formula, but that formula is not γ. In fact, γ becomes false in the process of being told and so it is logically impossible for you to know γ. Rather you learn, "there is a bug crawling on my shirt and I did not know it".

3 Intensional description of protocols

We have been considering systems given as sets of global histories. As we observed in the last section, while this stance suffices to make our central point about assigning meanings to messages, it may give rise to systems which are not realisable. As an example, consider a system where, for some $H \in E^\omega$, every prefix h of H is in $\mathcal{P}(\mathcal{H})$ but $H \notin \mathcal{H}$. Such a system cannot be realised by finite means, and indeed, one may wonder how such systems may even be presented. Thus reasonable descriptions of systems would place many closure conditions on $\mathcal{H}$.

Finite presentations (or specifications) of protocols are quite relevant for this discussion. If the meaning of a message may depend on the protocol used by agents, it is reasonable to ask how the protocol itself is given, so that such semantics may be 'computed'. Closely related to this is the question of what options an agent has at any point in time, and how the agent chooses one. Note that this is where *utility* and *informativity* of a message may play a critical role ([vR01]), particularly when the agent operates in an uncertain environment. In such a situation, the protocol determines the strategies available to an agent at any local state.

We first note that there are two distinct notions of protocols, both relevant, which are best explained by means of an example. When the notion of function was used in analysis, at first it meant simply a formula. E.g. polynomials and rational functions are easily expressed by means of formulae. Later this usage widened to include more general definitions and only with the advent of set theory do we have the most general notion of a function as a set of ordered pairs. The first two meanings are intensional, and the last is extensional.

If an individual process in a distributed computation needs to decide what its possibilities are for its next action, its decision can depend only on what it knows. If we already know which horse is going to win, before we place the bet, then life will be simple. Unfortunately that decision has to be made on the basis of what is known before the race is run. So clearly, if X is the set of possible next actions for some process i, with local history h, then X must depend functionally on h. The exception to this rule is a message received from the outside where the role of the process itself is passive.

In the context of distributed computing the message must have been sent by another process. Thus the value of the message is a function of the local history of that other process, and functionally dependent on the (current fragment of) global history. However in the context where a process or a group of processes are interacting with 'the world' about which they do not have complete information, they may still have some idea of the limitations on the set $\mathcal{H}$ of global histories. E.g. a young man receiving a letter from his fiancee might not know *what* the letter says or even *what protocol* if any she is following. Nonetheless he does know that the letter must have been mailed before receipt and hence cannot contain a reference to today's breaking news.

Thus an intensional description of a protocol is one that specifies, for each agent, at any finite global history, what global event that agent may participate in. This corresponds to the standard way protocols are described (informally but intuitively) in distributed algorithms, as a set of rules that specify when an agent may send a message, and what an agent must do on receipt of a message.

In fact, impossibility proofs like those in [FLP] and [LF] are really logical theorems about knowledge and the non-existence of intensional protocols such that the corresponding extensional protocol has some desired property. Similarly the unsolvability of the co-ordinated attack problem in the presence of asynchronous communication says that if an extensional protocol allows for the possibility of acquiring common knowledge, then the message mechanism in any corresponding intensional protocol cannot be asynchronous.

Returning to knowledge based semantics of messages, we have an apparent circularity when we consider intensionally presented protocols: the protocol talks of when messages be sent and what to do on receipt of a message, and the meaning of the message depends on the protocol used. This is resolved by considering only those extensional protocols as models of knowledge formulas that can be realised by intensional protocols. In this case, we can turn the problem around, and consider pairs of knowledge formulas to be *intensional specifications* of protocols.

For such an attempt to make sense, we need to demonstrate the following. If a knowledge formula has an extensional protocol as a model, then there is an intensional protocol which realises it. Moreover, we need to do this somehow for pairs of knowledge formulas that constitute semantic content of messages as defined earlier.

We first note that when (α, β) is in $Sem(i, m)$ for an agent i and event m, α

and β are global formulas but refer to successive **local** time instants for agent i. We therefore move to a logical framework where $\mathbf{K}_i\alpha$ and $\mathbf{K}_i\beta$ are local formulas when α and β are global formulas, and the $\bigcirc$ modality is also local, so that the pair (α, β) can be specified by a formula of the logic. We can then ask what properties must be satisfied by $Sem(i, m)$ so that we can actually construct, from the models, local 'choice' or 'strategy' functions for each agent. While we do not solve the problem here, we set up the framework so that the problem may be posed precisely. Rather than functions, we will present the agents as action labelled transition systems, which specify, for each agent, at any local state, what actions it can perform, based on its knowledge at that state.

We need some preliminaries, and some changes in definitions. A *distributed alphabet* is an n-tuple $\widetilde{\Sigma} = (\Sigma_1, \ldots, \Sigma_n)$, where for each $i \in [n]$, Σ_i is a finite nonempty alphabet of *i-actions* and for all $i \neq j$, $\Sigma_i \cap \Sigma_j = \emptyset$. $\Sigma = \bigcup_i \Sigma_i$ is the set of *system actions*, and we use a, b, c etc to refer to elements of Σ. The condition that $\Sigma_i \cap \Sigma_j = \emptyset$ for $i \neq j$, reflects the absence of synchronizations (or 'shared actions'), and hence a communication from i to j must be split into a send action in Σ_i and a receive action in Σ_j.

In the description of systems, we now give the states of agents explicitly (since these describe the information available to agents at different points in time), and derive the projection operations instead. Let $(Q_1, \ldots, Q_n)$ be an n-tuple, where Q_i is the set of possible local states of agent i. The product $\widetilde{Q} = (Q_1 \times \ldots \times Q_n)$ consists of possible global system states. We will use x, y etc. to denote global states, and the notation $x[i]$ to denote the i-local state in the i^{th} component of x. A **run** of the system is a sequence $\delta = x_0 a_0 x_1 a_1 \ldots$, where for $k \geq 0$, x_k is a global stste and $a_k \in \Sigma$ is a system action, such that, when $a_k \in \Sigma_i$, for all $j \neq i$, $x_k[j] = x_{k+1}[j]$. This legality condition ensures that only the agent participating in an action changes state.

Formally, a system over the distributed alphabet $\widetilde{\Sigma} = (\Sigma_1, \ldots, \Sigma_n)$ is a tuple $S = (Q_1, \ldots, Q_n, \mathcal{R})$, where $Q_1, \ldots, Q_n$ are the local states of agents in $[n]$ and $\mathcal{R}$ is a set of (infinite) runs of S. As before, we will consider $\mathcal{P}(\mathcal{R})$, the set of partial runs, which are finite prefixes of runs ending in system states.

Given a partial run δ, we can define a sequence $\lceil_i(\delta) \in Q_i^*$ by erasing all actions not in Σ_i and projecting down only i-local states. We can then define $\delta \sim_i \delta'$ iff $\lceil_i(\delta) = \lceil_i(\delta')$.

We will also change the syntax of the logic to talk of *local* knowledge assertions. Fix countable sets of *propositional letters* $(P_1, P_2, \ldots, P_n)$, where P_i consists of the atomic local properties of agent i. Let $P \stackrel{\text{def}}{=} \bigcup_i P_i$. Let $i \in [n]$. The syntax of i-local formulas is given below:

$$\Phi_i ::= p \in P_i \mid \neg \alpha \mid \alpha_1 \vee \alpha_2 \mid \bigcirc \alpha \mid \alpha_1 \mathbf{U} \alpha_2 \mid \mathbf{K}\, \psi,\ \psi \in \Psi$$

Global formulas are obtained by boolean combination of local formulas:

$$\Psi ::= \alpha@i,\ \alpha\ \in\ \Phi_i \mid \neg\ \psi \mid \psi_1\ \vee\ \psi_2$$

Thus the formulas are defined by mutual recursion, with i-local knowledge formulas refering to global properties, and global formulas being boolean combination of formulas of the form $\alpha@i$, where α is a local formula.

Models are now pairs of the form $M = (S, V)$, where $V : Q \to 2^P$ such that for $q \in Q_i$, $V(q) \subseteq P_i$.

Consider an infinite run $\delta \in \mathcal{R}$. Let $\rho = \lceil_i(\delta) = q_0 q_1 \ldots$; by ρ_k we denote the state q_k. The notion that an i-local formula α holds at local instant k for agent i in ρ is denoted $\rho, k \models_i \alpha$, and is defined inductively as usual:

- $\rho, k \models_i p$ iff $p \in V(\rho_k)$, for $p \in P_i$.
- $\rho, k \models_i \neg\alpha$ iff $\rho, k \not\models_i \alpha$.
- $\rho, k \models_i \alpha \vee \beta$ iff $\rho, k \models_i \alpha$ or $\rho, k \models_i \beta$.
- $\rho, k \models_i \bigcirc\alpha$ iff $\mid \rho \mid > k$ and $\rho, k+1 \models_i \alpha$.
- $\rho, k \models_i \alpha \mathbf{U} \beta$ iff there exists $m \geq k$ such that $\rho, m \models_i \beta$ and for all $\ell : k \leq \ell < m$, $\rho, \ell \models_i \alpha$.
- $\rho, k \models_i \mathbf{K}\psi$ iff for all $m \geq 0$, for all $\delta' \in \mathcal{R}$ such that $\lceil_i(\delta'_m) = q_0 q_1 \ldots q_k$, $\delta', m \models \psi$.

To define the semantics of global formulas, we need to relate global time instants with local time instants. For this, we extend the $\lceil_i$ map appropriately: $\lceil_i(\delta, 0) = 0$. $\lceil_i(\delta, k+1) = \lceil_i(\delta, k) + 1$ if $a_k \in \Sigma_i$ and $\lceil_i(\delta, k+1) = \lceil_i(\delta, k)$, otherwise.

- $\delta, k \models \alpha@i$ iff $\delta, \lceil_i(\delta, k) \models_i \alpha$.
- $\delta, k \models \neg\psi$ iff $\delta, k \not\models \psi$.
- $\delta, k \models \psi_1 \vee \psi_2$ iff $\delta, k \models \psi_1$ or $\delta, k \models \psi_2$.

Note that the i-local formula $\mathbf{K}\psi_1 \supset \bigcirc\mathbf{K}\psi_2$ carries the same semantics as a pair in $Sem(i, m)$ before, when $m \in \Sigma_i$.

We can now ask, given a formula ψ, if for every model $M = (S, V)$ of ψ, there exist transition systems $(Q_i, \to_i)$ such that $\mathcal{R}$ of S is obtained as the set of runs of a system obtained by (suitably) taking a product of these transition systems. When this question is answered positively, we have an intensional protocol in which messages carry the same meaning as those given by knowledge formulas as above.

In general, this is a hard question to answer. A partial answer to the question may be attempted using techniques that involve the theory of tree automata. It is not difficult to construct a nondeterministic Büchi tree automaton for every formula such that the models of the formula (in a technical sense) correspond exactly to the tree language accepted by the automaton. However, decomposing the tree automaton into individual transition systems for agents (thereby obtaining their 'strategies') seems to be difficult.

In the highly restricted context of *deterministic* systems, we can work with Büchi word automata, and in such situations, the required decomposition can also be done, along the lines of the exercise carried out in [R96a] and [R96b] for different logics. Such an exercise happily also yields finite transition systems for individual agents, thus demonstrating bounded memory intensional protocols. But then, in these systems, the agents know which run they are involved in, which is a very unrealistic assumption.

4 The amount of information in a message

Dretske [D] makes a convincing case that a signal (single symbol or a string) that may take one of n values can contain, on the average, at most $log(n)$ bits of information and cannot be used to distinguish between more than n possibilities. However, Dretske's arguments apply only to a very special situation where the signal is sent asynchronously between fixed parties. As we have seen before, in the context of synchronous communication, much more information may be conveyed. Thus we need a more general theory of the amount of information received by a process, which handles such cases satisfactorily.

We now consider situations where in intensional protocol, instead of just giving a finite set of possible extensions to the current global history, actually assigns probabilities to them. In this case we have a probabilistic protocol. For the purpose of the discussion, we again revert to the abstract model based on histories studied earlier. When H is global history and h is a local history of agent i, we write $Proj(H, i, h)$ to denote that there exists a k such that $h = \lceil_i(H_k)$.

We assume a probability measure on the set of all possible global histories with the following property. If H is a history in the protocol, then for every k, the measure $m(\{H' \mid H'_k = H_k\})$ is positive. If the protocol arises from a probabilistic intensional protocol, then such a measure will arise naturally from the probabilities of local events at various moments. Otherwise we stipulate this as a condition on the protocol. The assumption of positive measure is not strictly speaking necessary. We will be using conditional probabilities. and it is shown in [PP1], [PP2] that using non-standard analysis we can define conditional probabilities $p(X/Y)$ even when both X and Y have zero measure. However if X and Y both have positive measure, then $p(X/Y)$ is simply the measure $m(X \cap Y/m(Y))$.

We now define the **amount** of information $inf(h.m)$ contained in a signal m for process i when its local history is h. It is $-log(p(X/Y))$, where Y is the set of all

global histories such that $Proj(H, i, h)$ and X is the set of all global histories such that $Proj(H, i, h; m)$.

Theorem 4.1 *Let A be a formula such that at history h, $A, \neg A$ are equiprobable to process i and m and m' are messages such that if process i receives m, it will know that A and if it receives m', it will know that $\neg A$. Then the average amount of information in m, m' at h, $(inf(h, m) + inf(h.m'))/2$, is at at least 1 bit. More generally, if $A_1, \ldots, A_k$ are mutually exclusive formulas which are equi-probable to i at h, and signals $m_1, \ldots, m_k$, will respectively result in i knowing $A_1, \ldots, A_k$, then the average information in the signals at h is at at least $log(k)$.*

This fact is well known to information theorists and we skip the proof.

Theorem 4.2 *If i does not know A at h and the subjective probability of A for i is p, then the minimum amount of information in any signal m that results in i knowing A is $-log(p)$.*

This is because the set of histories in which the signal m has been received is a subset of the set of histories in which A is true and the latter has probability p.

Here the subjective probability of A for i at h is

$$m(\{H \mid H \models A \text{ and } Proj(H, i, h)\})/m(\{H \mid Proj(H, i, h)\})$$

Note that none of the development above really depends on the assumption that m is a message coming from outside. m can be replaced by an addition h' to history h and $inf(h, h')$ is the total information received by i between the local histories h and $h; h'$. We can also define the amount of information received at the history h, $I(h) = inf(\emptyset; h)$, where $\emptyset$ is the empty history.

Theorem 4.3 *$I(hh') = I(h) + inf(h, h')$. I.e. the information received between h and h' plus that received from the start to h, equals that from start to hh'.*

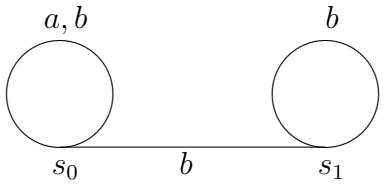
Similarly we can define the information $I(i, j, h)$ received by i **about** j's knowledge, during history h and it can be readily shown that:

Theorem 4.4 *$I(i, j, h) \leq I(h)$.*

It is easily seen that if the process i is deterministic, then all the information it gets must come from outside. Moreover, if the process is itself probabilistic, then it gets information from itself too (or from its own coin, if you like). This is also true in the nondeterministic case, though there we do not have a measure of the **amount** of information.

We illustrate this by means of an example. Suppose there are two processes 1 and 2 which send each other messages. 1 sends a symbol which may be either a or b, and on receipt, 2 sends a message which may be c or d, and so on. Then the e-protocol is simply $((a+b)(c+d))^\omega$. Now suppose that at some moment 1 decides to send only a's from now on. It then knows the formula, "there are only finitely many b's". But it could not have learned this formula from its own local history which is finite. How do we resolve this ?

Suppose in fact that the process i is a nondeterministic finite automaton which has a state diagram (somewhat simplified) as below:



where s_0 is the initial state. If after the exchange bab it decides to send only b's from now on, then its local history is not bab, but $s_0bas_0bs_1$ and the last s_1 contains the needed information. Thus the process 1 did not know in **advance** that it was going to go into state s_1, and it did not **decide** to go into state s_1. Rather, that decision was made from outside, by the scheduler, and it conveyed information to the process, just as it does to us. Of course if the automaton were deterministic, then such acquisition of information could not take place.

To show that issues here are complex, we now describe an example given to us by Bill Gasarch. Let W be an undecidable r.e. set and we are given four numbers m, n, p, q. We are allowed to ask God two questions and then we must decide which of the four numbers are in W. At first sight this seems impossible as there are sixteen possibilities, and we are allowed only two questions. However, there is a solution. [1]

5 Understanding the Lion

There is some dispute about the exact import of what Wittgenstein meant by the remark we have cited at the beginning of the paper. But there is an obvious sense in which we *do* understand lions. If a lion growls, it very likely means "Scram!" and surely does not mean, "I saw an elephant this morning". So we do have some understanding of lion language.

This can be easily modelled in our formal discussion. Note that when we defined $Sem(i, m)$ as a set of pairs of formulas, there is no need whatever to presume that the language from which elements of $Sem(i, m)$ are taken needs to be the same as the

[1] We use the two questions to find out from God how many of the four numbers are in W. After we get the answer, say 2, we enumerate W (it is r.e.) until two of the four numbers have turned up. At this point we know that the other two will not turn up.

language from which elements of $Sem(j, m) : i \neq j$ are taken. It is perfectly possible for the languages of i, j to be private (syntactically) and yet the signal m conveys the same meaning (as a set of possible global histories) to both sender and receiver. [2] Thus if the lashing of tail (ℓ) by a lion is always followed by a charge c whereas growling g is not, then in all global histories an ℓ event will be followed by a c event and the formula pair $(True, C)$ will be in $Sem(i, \ell)$ but not in $Sem(i, g)$ where C stands for "The lion will soon charge". i of course is us, the intended targets of the charge.

We have a similar explanation for the arrangement between Paul Revere and his friend. Let ℓ mean that the British come by land and s that the come by sea. Let o be the event that one lantern is shown and t the event that two are shown. By arrangement all sufficiently long global histories contain the sequence ℓ, o or else the sequence s, t. They never contain ℓ, t or s, o. Now ℓ, s are local events for the friend and o, t for Paul Revere. Thus if L is the formula which is true when the British come by land and S the formula which is true if the British come by sea, then the formulas $(True, L)$ is in $Sem(p, o)$ and $(True, S)$ in $Sem(p, t)$.

When a Hindi speaker says *jaldi!*, after some experience an English speaker will figure out that what is meant is *quickly!*. This is even more clear with a more colorful word like *phata-phut!* which has the same meaning. When i, j are communicating, ultimately the languages of both i and j describe sets of global histories and provided that a formula A in $i's$ language is reliably followed by signal m, and expresses a set of global histories X, j can, under favourable conditions figure out that m 'means' X and need never know what A was. If B is *j's* word for X then j will reliably interpret m as meaning B.

The use of "Rain in Spain is mainly in the plain" can also be accomodated into the history semantics. Let G be a group of conspirators, all of whom are informed by the leader: "for this evening, our code sentence will be "Rain in Spain is mainly in the plain". Now the utterance of this sentence in the absence of any context is highly unlikely. Thus if I am a member of the group and hear you say "Rain in Spain is mainly in the plain", then with high probability, your own local history includes an event of hearing the communication from the leader and hence that you are a member of the group. (Of course moles cannot be revealed this way). It is crucial that the probability of hearing a code sentence accidentally should be low and hence a sentence like "It is terrible that the US is planning to invade Iraq" cannot be used as a code sentence.

[2] See [P94b] for a discussion of the case where because of vagueness the meanings are not the same, but the signal does, nonetheless, result in an increase of utility measured in terms of the time taken by an algorithm.

6 Meaning and Implicature

In [Gri89] Paul Grice introduces the notion of *implicature* [3] where what is implicated in an utterance is *more* than what is actually said. For instance the question "Do you have any salt?" normally carries the implicature, "I'd like some". How does implicature differ from plain meaning?

This difference can also be explained in our framework. We defined $Sem(i, m)$ to be $\{(\alpha, \beta) \mid$ for all $H \in \mathcal{H}$, for all $k \geq 0$, if $H, k \models \mathbf{K}_i\alpha$ then, for all $h' \in \mathcal{P}(\mathcal{H})$ such that $H_k \preceq h'$, h' is of length $\ell \geq k$ and $\lceil_i(h') = \lceil_i(H_k)m$, we have $H', \ell \models \mathbf{K}_i\beta\}$. This is the meaning of m for process i. But we can make this notion both wider and narrower. Wider in the sense that we can remove i as a parameter and demand that the meaning apply to *all* processes. This will be the semantic meaning of m *simpliciter*. On the other hand, in restricted circumstances we may have *more* pairs. Let us define $Sem(h, i, m)$ to be $\{(\alpha, \beta) \mid$ for all $H \in \mathcal{H}$, for all $k \geq 0$, if $H, k \models \mathbf{K}_i\alpha$ and $\lceil_i(H) =_k h$ then, for all $H' \in \mathcal{P}(\mathcal{H})$ such that $H_k \preceq H'$, H' is of length $\ell > k$ and $\lceil_i(H') = \lceil_i(H_k)m$, we have $H', \ell \models \mathbf{K}_i\beta\}$.

Now $Sem(h, i, m)$ is likely to contain more pairs than $Sem(i, m)$ since we are restricting ourselves to histories H such that $\lceil(H_k) = h$ for some k. Thus we can identify $Sem(m)$ with the (conventional) meaning of m, $Sem(i, m)$ with the meaning of m for i, and $Sem(h, i, m)$ to be the meaning of m for i *in the context* h. In general, $Sem(m) \subseteq Sem(i, m) \subseteq Sem(h, i, m)$.

To use Grice's example, if h is a history which culminated with your car running out of gas, m is the message that there is a gas station around the corner, and B is the formula which is true iff the gas station is open, then the formula pair $(True, B)$ will be in $Sem(h, i, m)$ but not in $Sem(m)$. For example if I were renting an apartment and wanted to know if I would be able to get gas for my car, then the statement that there is a gas station around the corner would not carry the implicature that the gas station was open *at the time* that the statement was made. However, it *would* carry the implicature that it is open *some* time. The various implicatures do hold only because we are assuming that the histories contain (only) sequences of co-operative communications satisfying Grice's requirements.

References

[CM] M. Chandy and J. Misra, "How processes learn". *ACM PODC 1985*, pp 204-214.

[D] F. Dretske, *Knowledge and the flow of information*, MIT Press, 1981.

[FLP] M. Fischer, N. Lynch and M. Paterson. "Impossibility of distributed consensus with one faulty process", in *ACM PODS 1983*, pp 1-7.

[3] Actually it was introduced prior to the 1989 book which is merely a collection of his papers.

[FHMV95] Fagin, R., Halpern, J., Moses, Y. and Vardi, M., *Reasoning about knowledge*, M.I.T. Press, 1995.

[Gri89] Grice, P., *Studies in the Way of Words*, Harvard University Press, 1989.

[HF] J. Halpern and R. Fagin, "A formal model of knowledge, action and communication in a distributed system", in *ACM PODC 1985*, pp 224-236.

[HM] J. Halpern and Y. Moses, "Knowledge and common knowledge in a distributed environment", in *ACM PODC 1984*, pp 50-61.

[HMV94] Halpern, J., Moses, Y., and Vardi, M., "Algorithmic knowledge", *TARK V*, Theoretical Aspects of Reasoning about Knowledge, 1994, 255-266.

[KPN90] Krasucki, P., Parikh, R., and Ndjatou, N., "Probabilistic Knowledge and Probabilistic Common Knowledge" *ISMIS 90*, North Holland 1990, pp. 1-8.

[LF] N. Lynch and M. Fischer. "A lower bound for the time to assure interactive consistency", *Inf. Proc. Letters*. vol 14 (1982).

[Pa] Parikh, R., "Monotonic and Non-monotonic Logics of Knowledge", in *Fundamenta Informatica* special issue, *Logics for Artificial Intelligence* vol XV (1991) pp. 255-274 (appeared previously as "Logics of knowledge, games and dynamic logic", *Proc. 4th FST/TCS*, LNCS #181, pp 202-222.)

[P94] Parikh, R., "Logical omniscience" in *Logic and Computational Complexity*, LNCS 960, 22-29.

[P94b] Parikh., R., "Vagueness and Utility: the Semantics of Common Nouns" in *Linguistics and Philosophy* **17** 1994, 521-35.

[P95] Parikh, R., "Knowledge based computation (Extended abstract)" in *Proceedings of AMAST-95* Montreal, July 1995, Edited by Alagar and Nivat, LNCS no. 936, 127-42.

[PP1] Parikh, R., and Parnes, M., "Conditional probability can be defined for arbitrary pairs of sets of reals", *Advances in Mathematics*, 9 (1972), pp 313-315.

[PP2] Parikh, R., and Parnes, M.,"Conditional probabilities and uniform sets", *Proc. Victoria Symp. on Nonstandard Analysis*, Lecture Notes in Mathematics #369, pp 180-194.

[PR86] Parikh, R., Ramanujam, R., "Knowledge, information and protocols (extended abstract)", manuscript, 1986.

[R96a] Ramanujam, R., "Local knowledge assertions in a changing world", *Proc TARK VI*, Theoretical Aspects of Rationality and Knowledge, 1996, 1-17.

[R96b] Ramanujam, R., "Locally linear time temporal logic", *Proc LICS*, 1996, 118-127.

[vR01] van Rooy, R., "Utility, informativity and protocols", research report, 2001.

[Sk96] Skyrms, B., *Evolution of the Social Contract*, Cambridge University Press 1996.

[T94] Tel, G., *Introduction to distributed algorithms*, Cambridge University Press, 1994.