Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

193

Logics of Programs

Brooklyn, June 1985

Edited by Rohit Parikh

Springer-Verlag
Berlin Heidelberg New York Tokyo

Distributed Processes and the Logic of Knowledge
(Preliminary Report)

Rohit Parikh[1] and R. Ramanujam[2]

Abstract: We establish a very natural connection between distributed processes and the logic of knowledge which promises to shed light on both areas.

Introduction: Interest in the logic of knowledge among computer scientists has recently taken a sharp upturn as evidenced by a number of recent publications [XW], [FHV], [HM], [Le], [Pa]. In particular the last three papers emphasise the possibility of a connection between the logic of knowledge and problems in distributed processing, for example between questions about Byzantine general problems and the possibility of attaining common knowledge under specified circumstances. In this paper we carry out a somewhat more systematic analysis, giving a definition of what it means for an individual process in a distributed system to know some fact and what axioms this knowledge obeys.

Philosophers have traditionally defined knowledge as justified true belief, at least until recently, when problems with this definition were pointed out by Gettier [G]. Nonetheless, a consensus remains that knowledge at least requires true belief and it is not clear what meaning we may give to the "beliefs" of an individual process in a distributed system.

Another problem for the logic of knowledge, at least as applied to humans, is that all prevalent logics of

---

1. Department of Computer Science, Brooklyn College and Mathematics Department, CUNY Graduate center. Research supported in part by NSF grant MCS83-04959 and a Faculty Research Assistance grant from the research foundation of CUNY.

2. Computer science group, Tata Institute of Fundamental Research, Homi Bhabha Road, Bombay 400005, India.

knowledge assume that the knowledge of any particular person is closed under logical inference. Humans are notorious, however, for failing to carry out all the logical inferences that are available to them. Thus any existing logic of knowledge as applied to humans is at best an approximation. A similar problem arises in public key cryptography since the public key together with the coded message do contain the decoded message as information, and the available logics are unable to adequately represent the fact that the cost of actually extracting this information is prohibitive. In other words, existing logics are able to represent a lack of knowledge, when this lack arises due to the absence of some necessary information; but they are unable to represent a lack of knowledge that arises when an inference which is possible, is not in fact carried out.

It will turn out, as we shall see below, that this problem does not arise with our definition and that individual processes do in fact know just what they ought to know. This is because the definition is purely information based and does not involve anything that could be called introspection.

The Basic Notions: In the following we assume that we have n processes $1 \leq i \leq n$ which may communicate with each other along binary channels, by broadcast, limited or global, or by setting variables to which more than one process may have access. The set I will be the set of all processes. We assume a global clock at the metalevel, i.e. that all events are partially ordered by a global (discrete) time, (two events may take place simultaneously) but we do not assume, except where explicitly stated, that the processes themselves have access to this or any other clock.

By a local history we shall mean the sequence of all (relevant) events, in the order of occurrence, internal events as well as messages sent and received, for some process i. In case there is a common clock, then the events will be time stamped, i.e. each element of the local history will be a pair (t,e) where e is the event and t is the instant of time at which the event takes place. Variables h, h' etc will range over local histories. By considering compound events we may assume that each process has at most one event happening at any moment of global time.

By a global history we shall mean a sequence of pairs (t, X) where t is the global time and X is some set of local events. As said before, we may assume that at most one event of some particular process i is in X. Variables H, H' etc will range over global histories.

We define a map Φ_i from global events to local events$\cup$(null) as follows.

$\Phi_i(t, X)$ = null if there is no i-event in X.

otherwise

$\Phi_i(t, X)$ = the unique i-event in X, if there is no common clock,

= (t, e) when there is a common clock and e is the unique i-event in X.

Φ_i extends to a unique map (also called Φ_i) which preserves concatenation and whose domain is the set of global histories. If U is a subset of I then $\Phi_U(H)$ is the sequence $\langle \Phi_i(H) : i \in U \rangle$. θ is Φ_I.

Clearly the map θ is one to one if there is a common clock or what amounts to the same thing, some event is happening at every individual process at every moment of global time. In the absence of a common clock, a weaker property may sometimes hold.

Call two global histories H, H' equivalent if the same sequence of events is obtained when the time stamps are erased.

Many properties of interest do not distinguish between equivalent histories. Examples are fairness (of various kinds) and deadlock freedom (provided, as is usually the case, that the boolean guards controlling parallelism do not depend explicitly on time, but only on the values of certain program variables.)

Then the map θ will be weakly one to one if $\theta(H) = \theta(H')$ implies that H and H' are equivalent. An example

where θ is weakly one to one but not one to one, would be a blind chess game played by correspondence, in which case the players need not possess a clock nor need they know the moves of the other player, but the game can be uniquely determined if the moves of the white and black players are given separately.

There is a map TL which forgets time stamps and converts global histories into timeless histories, i.e. sequences of sets of events. Evidently, H, H' are equivalent iff $TL(H)=TL(H')$. By a protocol we shall mean simply a set of possible global histories closed under the "prefix of" relation. (We shall write $H \leq H'$ to mean that H is a prefix of H'.) There may be some finite mechanism for ensuring that no history outside the protocol is actually ever possible and usually it is this mechanism that is called the protocol, but we shall prefer to work abstractly and hope that the reader will forgive this transgession. This decision is really analogous to the decision in mathematics to study sets rather than properties. To use another analogy, if the usual notion of a protocol corresponds to a grammar, then ours corresponds to the language generated by that grammar. If we are interested in fairness, then the protocol will consist only of fair histories. There is of course no finite mechanism to achieve exactly the fair histories.

A protocol P is weakly one to one if for all H, H' in P, $\theta(H)=\theta(H')$ implies $TL(H)=TL(H')$. It is full if $H \in P$ and $TL(H')=TL(H)$ imply $H' \in P$. If a protocol is not weakly one to one, then there are likely to be questions about the current global history of the system which cannot be answered even by pooling the knowledge of all the systems. Problems that depend on such knowledge being available will then not be soluble.

We assume some language L whose formulas correspond to properties of individual global histories. Such properties may be, "the local variable x is now zero", "the variable y has never been zero" or protocol dependent properties like "this history has no proper extensions in the protocol". L will be assumed to have the classical boolean connectives. L will be time-free, if for all H, H' and $A \in L$, $TL(H)=TL(H')$ and $H \models A$ implies $H' \models A$. L will be separating if for all H, H' with $TL(H) \neq TL(H')$ there is an A in L such that $H \models A$ and $H' \models \neg A$.

We extend L to a language L^+ as follows.

(i) Every A in L is in L^+.

(ii) If A, B are in L^+, so are $\neg A$ and $A \vee B$.

(iii) If A is in L^+ and U is a set of processes, the $K_U(A)$ is in L^+.

(iv) If A is in L^+ then so are GA, FA, AUB.

We now define the relationship $H \models A$ for H in the protocol P and A in L^+ as follows.

(i) If $A \in L$ then the notion is supposedly given as part of the definition of L.

(ii) $H \models \neg A$ iff it is not the case that $H \models A$. Similarly for $A \vee B$.

(iii) $H \models K_U(A)$ iff for all H' with $\Phi_U(H) = \Phi_U(H')$, $H' \models A$.

(iv) $H \models GA$ iff all H' such that $H \leq H'$, $H' \models A$.

(v) $H \models FA$ iff for all maximal sequences $H_1, \ldots H_i, \ldots$ with $H = H_1$ and $H_i < H_{i+1}$ for all i, there is an i such that $H_i \models A$.

(vi) $H \models AUB$ iff for all maximal sequences $H_1, \ldots$ as above, there is an $i > 1$ such that $H_i \models B$ and for all j, $1 < j < i$, $H_j \models A$.

Note that the construct K_U does not represent common knowledge, but is in some sense dual. It is straightforward to show that $K_{\{1,2\}}$ cannot be expressed in terms of K_1 and K_2.

Intuitively $K_{\{1,2\}}(A)$ means, 1 and 2 together have enough information to deduce A.

Note that the connectives F and X (next) can be defined from U as follows. FA is trueUA where true can be any simple tautology. Similarly XA, (next A) can be defined as falseUA.

Note also that that knowledge is really dependent on three parameters. The individual i, the fact A, and finally, the protocol P. We do not assume that the protocol is known to the processes in any sense. Rather, it is background knowledge of the programmer. Thus for example, suppose the programmer knows that given the program, if gcd(x, y) = z if the boolean flag f is true, and that action α is safe if gcd(x, y) = z then he can perfectly well program a process to do α just in case f is true. The process does not need to know why f contains the right information, only the programmer needs to know that.

Finally, we point out that we have defined two global histories H, H' to be equivalent for i iff $\Phi_i(H)=\Phi_i(H')$. (similarly for a subset U of I) However, in practice a a more severe equivalence relation may be relevant. I.e. given the initial state s of a process and a local history h, there is a new state t depending only on s and h, let us call it t=h(s). Then one would really like to say that knowledge of i depends not on h, but only on h(s) and that if h(s)=h'(s) but $h\neq h'$, then the information that would distinguish between h and h', though i once had it, is now forgotten and not available to the program. It is clear that this is the right intuition for some applications, e.g. if i is a finite automaton, then the truth of theorems like the pumping lemma depends on the fact that i is awfully forgetful so that things that he knew at one time, are now gone from his memory.

The Basic Results:

Theorem 1: If L is time free and P is full, then L^+ is also time free. If, moreover, P is weakly one one, then $A\Leftrightarrow K_i(A)$ is valid in the model (holds for all histories). If L is separating and $A\Leftrightarrow K_i(A)$ is valid, then P is weakly one one.

Proof: Recall that H, H' are equvalent if TL(H)=TL(H'). We need to show that for all H, H' equivalent and all A in L^+, $H\models A$ iff $H'\models A$. This is obvious for elements of L and is preserved under the truth functional connectives. Now note that if H, H' are equivalent, and U is a subset of I, then $\Phi_U(H)=\Phi_U(H')$. Hence if for all H'' with $\Phi_U(H'')=\Phi_U(H)$ implies $H''\models A$, then for all H'' with $\Phi_U(H'')=\Phi_U(H')$ also implies $H''\models A$. I.e. $H'\models K_U(A)$ implies $H\models K_U(A)$ and vice versa. This settles the case for K_U.

To deal with the connectives G, F, and U we need the fullness of P. Suppose for example that H, H' are equivalent. Then if $H\models GA$ does not hold, then there is an H'' extending H such that $H''\models\neg A$. But then there is an H''' which extends H' and such that TL(H''')=TL(H''). By fullness H''' is in P. Hence H' does not satisfy GA either. The other cases are similar.

It immediately follows that if P is weakly one to one, then A and $K_i(A)$ must be equivalent, for $\Phi_i(H)=\Phi_i(H')$ implies TL(H)=TL(H'). Thus if $H\models A$, then we must have $H\models K_i(A)$.

Conversely, if L is separating and P is not weakly one to one, then there are H, H' with $TL(H) \neq TL(H')$ but $\Phi_I(H) = \Phi_I(H')$. Now there is a formula A of L such that $H \vdash A$ and $H' \nvdash A$. Then $H \nvdash K_I(A)$ so that A and $K_I(A)$ are not equivalent. ∎

In the following theorem we consider only the connectives F, G, and X.

Theorem 2: The following axiom system is sound.

(i) Tautologies

(ii) $K_I(A) \Rightarrow A$

(iii) For $U \subseteq V$, $K_U(A) \Rightarrow K_V(A)$

(iv) $K_U(A) \Rightarrow \{K_U(A \Rightarrow B) \Rightarrow K_U(B)\}$

(v) $K_U(A) \Rightarrow K_U(K_U(A))$

(vi) $\neg K_U(A) \Rightarrow K_U(\neg K_U(A))$

(vii) $GA \Rightarrow (A \& GGA)$

(viii) $K_U(GA) \Rightarrow G(K_U(GA))$

(ix) $FA \& G(XA \Rightarrow A) \Rightarrow A$

(x) $FA \Leftrightarrow A \vee XFA$

Rules of inference

$$\frac{A \quad A \Rightarrow B}{B}$$ (modus ponens)

$$\frac{A \Rightarrow B}{K_U(A) \Rightarrow K_U(B)}$$ (K-generalisation)

(similarly for F and X which are both monotonic)

$$\frac{A \Rightarrow XA}{A \Rightarrow GA}$$

There cannot be a completeness result since the details of L are not known. L may, for instance, have quantifiers. We suspect that in case knowledge depends only on the current state of a process and not on its total history, and there are only finitely many states for each process, then the logic will be decidable by reduction to CTL* or SnS. See [ES], [HKP].

Proof: Most of the proofs are quite straightforward. Axioms (i) are obvious. Axioms (ii), (iv), (v) follows from the fact that the relation of being U-equivalent $\Phi_U(H)=\Phi_U(H')$ among histories is an equivalence relation and that if U is a subset of V, then V-equivalence implies U-equivalence. ∎

In the following theorem, we make an assumption about the protocol which is almost always realised in practice. Namely, that what a process i is about to do next depends only on its history h so far and not directly on the current state of any of the other processes. (Of course there may be an indirect dependence due to a prior communication between i and another process) We assume moreover that the relative speeds of the processes, though finite, are undetermined. Thus let $h_i \to h_i'$ mean that there are global histories H, H' with $H<H'$, $\Phi_i(H)=h_i$, $\Phi_i(H')=h_i'$ and h_i' is an immediate proper extenson of h_i. (H' need not be an immediate extension of H) Suppose now that we are in a state where the local hostory of i is h_i, $h_i \to h_i'$, h_j is the current local history of j and $h_j \to h_j'$. Then all of the next pairs of local histories are possible for i, j: (h_i, h_j'), (h_i', h_j) and (h_i', h_j'). I.e. one or both processes may go ahead to the next step. A similar remark will apply if more than two processes are involved.

Theorem 3: Let P be a protocol for several processes to share a critical section (CS). Safety and liveness properties of the protocol are related to the knowledge properties in the following way.

(i) The protocol is safe in the sense that two systems cannot simultaneously enter the CS iff no process attempts to enter the CS unless it "knows" that it is empty and that no other process is about to enter it.

(ii) If the protocol is safe and every process must terminate properly, then whenever the CS is empty and some process needs to enter it, then eventually it comes to know that the CS is empty.

Proof: (i) Suppose that the current global history is H, process i is about to enter the CS (i.e. h_i has an immediate extension where entering the CS was the last step) H' is any history i-equivalent to H, and H" is an immediate extension of H'. We must show that no other process can be in the CS in H' or H". For suppose some other

process j was in the CS in H'. $\Phi_j(H')=h_j$ with j in CS and $\Phi_i(H')=\Phi_i(H) = h_i$ (say) and h_i has an immediate extension h_i' where i has entered the CS. Then the pair (h_i', h_j) is in the protocol which was not safe after all. A similar argument applies if j was not in the CS but was about to enter it. Since H' was arbitrary with $\Phi_i(H')=\Phi_i(H)$, then process i knows that the CS is empty etc.

We also have to show that if every process knows, whenever it is about to enter the CS that it is safe to do so, then it is obvious that the protocol itself is safe, since no global history can have two processes in the CS simultaneously.

(ii) Suppose that process i needs to enter the CS, then since we know that eventually i must enter the CS, by part (i) eventually i must come to know that the CS is available. ∎

The theorem above shows that for a protocol to guarantee safety and liveness, it must provide for an exchange of information in a a systematic way so that the processes can acquire the knowledge that they need.

We illustrate by the following simple example.

Consider two processes 1 and 2 which share a CS. There is a boolean variable x which process 1 can set and which process 2 can read. Similarly with y and 1,2 interchanged. When the process 1 wants to enter the CS it proceeds as follows:

(i) Set x to a random value. If x=1 then goto (i).
(ii) test y. If y=0 then goto (i).
(iii) Enter the critical section.
(iv) Set x to 1.

The procedure for process 2 is dual.

We can show by informal arguments that the protocol is safe (and also live with probability 1). However, let us consider the safety from the point of view of knowledge.

Now suppose 1 is about to enter the CS. Then its local history h_1 ends with the steps x:=0; y=1?. If process 2 were in the CS or about to enter the CS then its history h_2 would end y:=0; x=1? or y:=0; x=1?; CS2. However, no global history H has the property that $\Phi_1(H)=h_1$ and $\Phi_2(H)=h_2$. Thus process 1 knows that the CS is empty and that 2 is not about to enter it.

Common Knowledge: Let E(A) denote $\bigwedge K_i(A): i\in I$. Note that E(A) is stronger than $K_i(A)$ whereas $K_i(A)$ is weaker. E(A) means, everyone knows A. Then C(A), A is common knowledge, stands for the infinite conjunction $E(A) \wedge E(E(A)) \wedge \ldots$. We can define $C_U(A)$ for any subset $U \subseteq I$ by replacing I by U in the definition above. Common knowledge has been investigated by both [HM] and [Le]. Our results do not always agree with those in [HM] but it canbe verified that ours do hold for the definition we have given.

Consider a stable statement A, i.e. one which implies GA. Then it is straightforward to show that if A ever becomes common knowledge, then it does so simultaneously. More specifically, let H be a minimal history such that $H \models C_U(A)$. Then unless H is the empty history, H is of the form H';(t,X) where X has an event from every process in U. Moreover, if $\Phi_i(H)=h_i$ where $i\in U$, then h_i is minimal such that for all H' with $\Phi_i(H')=h_i$, $H' \models C_U(A)$.

Note however that attaining common knowledge depends on the protocol P. One can attain more common knowledge in a "narrower" P than in a wider one.

Using the observation above we can derive the following somewhat paradoxical result. Suppose that A and B are two friends, A lives in New York, B in Boston. The mail from New York to Boston takes exactly four days. On November 1, A writes to B that he has a new apartment, giving his address, Though he does not date his letter, the date can be guessed when it arrives on November 5, and it can easily be verified that A's new address becomes common knowledge on that date.

There is then an improvement in the mail service, and mail starts arriving faster, taking either two or three days. On January 10, A writes to B again that he has a new dog. Let D be the proposition that A has a dog. It turns out that D will never become common knowledge!

The intuitive reason is as follows. suppose in fact that the letter arrives on January 13. On the 13th both A and B know D. However B does not know that A knows that B knows D. For if the letter was received on the 13th, it might have been mailed on the 11th, and then A could not be sure of receipt till the 14th. Thus on the 14th $K_B(K_A(K_B(D)))$ is true. However, even then, $K_B(K_A(K_B(K_A(K_B(D)))))$ does not yet hold. For if the letter was mailed on the 11th, true as far as B knows, it might have, from A's point of view, been received only on the 14th in which case B would have to allow for the possibility (in A's mind) that the letter might have been mailed on the 12th and so...

To see this formally, consider histories H_j defined as follows: H_1 is the actual history where the letter is mailed on January 10 and arrives on the 13th. For even numbers $2j$, H_{2j} is the history where the letter is mailed on January 10+j and arrives on January 12+j, i.e. two days later. For odd numbers $2j+1$, the history H_{2j+1} is that where the letter is mailed on January 10+j and arrives on January 13+j.

It is easily checked that $\Phi_1(H_{2j})=\Phi_1(H_{2j+1})$, and that $\Phi_2(H_{2j-1})=\Phi_2(H_{2j})$. After j days have passed, the formula $K_A(K_BK_A)^j(K_B(D))$ is not yet true, since the real history H_1 is linked to the history H_{2j+1} by a chain of length 2j, and in that history B has not yet heard about the dog!

If the letter does arrive by chance on the 12th, the situation does not change much since of course A does not know this and a similar argument applies. Similarly if B acknowledges receipt of the letter, provided he does not date his reply.

However, things are not too bad. The formula $E(D)$ holds on the 13th, $E(E(D))$ on the 14th and so on. Hence the level of knowledge is rising as time passes even though there is no further communication. Now it was shown in [Pa], page 210 that if there is any formula B which does not have the construct C, and B is implied by $C(D)$, then it is also implied by some $E^j(D)$ and hence will be true after a finite number of days.

We finally point out that the fact that D is (reasonably) persistent is relevant to the discussion. If A were to write "it is snowing today", then it does not make sense to say that that fact could be common knowledge, though the persistent propositions, "A has experienced snow" or "it snowed in New York on January 10" can of course be under suitable circumstances.

Conclusions: The results established here are only preliminary and the real purpose is to show that the connection has genuine possibilities. Obvious extensions of the current work include a completeness result corresponding to theorem 2 when the ground language L is simple; handling the probabilistic case where there is a probabilistic distribution on the protocol P; and finally, if some aspects of the structure of the system are hidden from some of the processes, then allowing for different processes to "imagine" different protocols P_i.

References

[ES] E. Emerson and A. Sistla, "Deciding Branching Time Logic", STOC 84 14-24.

[FHV] R. Fagin, J. Halpern and M. Vardi, "A Model Theoretic Analysis of Knowledge", IEEE-FOCS 1984, 268-278.

[G] E. Gettier "Is Justified True Belief Knowledge?", Analysis 1963, 121-123.

[Hi] J. Hintikka, Knowledge and Belief, Cornell University Press 1962.

[HM] J. Halpern and Y. Moses, "Knowledge and Common Knowledge in a Distributed Environment", ACM-PODC 1984, 50-61.

[HKP] D. Harel, D. Kozen and R. Parikh, "Process Logic: Expressiveness, Decidability, Completeness", JCSS 25 (1982) 144-170.

[Ho] C.A.R. Hoare, "Communicating Sequential Processes", CACM, 21 (1978) 666-677.

[Le] D. Lehmann, "Knowledge, Common Knowledge and Related Puzzles", ACM-PODC 1984, 62-67.

[MSHI] J. McCarthy, M. Sato, T. Hayashi, S. Igarashi, "On the Model Theory of Knowledge", Computer Science Technical Report, Stanford 1978.

[Pa] R. Parikh, "Logics of Knowledge, Games and Non-monotonic Logic", FST-TCS 1984. Springer LNCS #181, 202-222.

[S] M. Sato, "A Study of Kripke-type Models of some Modal Logics by Gentzen's sequential Method" Pub. Res. Inst. Math. Sci. Kyoto University, 1977.

[XW] M. Xiwen and G. Welde, "A Modal Logic of Knowledge", IJCAI 1983, 398-401.