FOUNDATIONS OF KNOWLEDGE FOR DISTRIBUTED SYSTEMS

Michael J. Fischer and Neil Immerman

YALEU/DCS/TR 426
September, 1985

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 426	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) FOUNDATIONS OF KNOWLEDGE FOR DISTRIBUTED SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(*s*) Michael J. Fischer and Neil Immerman		8. CONTRACT OR GRANT NUMBER(*s*) NSF: DCR-8405478 and ONR: N00014-82-K-0154
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Computer Science/ Yale University Dunham Lab/ 10 Hillhouse Avenue New Haven, Connecticut 06520		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research 800 N. Quincy Arlington, Virginia 22217		12. REPORT DATE September, 1985
		13. NUMBER OF PAGES 11
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)		15. SECURITY CLASS. (*of this report*) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distributed unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

knowledge
common knowledge
distributed system
protocol
logic
formal model
foundations

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

General and precise definitions are given of distributed protocol and of various notions of knowledge and common knowledge in distributed systems. Whether or not a process in a distributed protocol can obtain new common knowledge during execution is shown to depend on which notion of common knowledge chosen, showing that the problem of capturing intuitive concepts about knowledge is more subtle than was previously believed.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

Foundations of Knowledge for Distributed Systems

Michael J. Fischer* and Neil Immerman†

Computer Science Department
Yale University
New Haven, CT 06520

September 19, 1985

1 Introduction

In [HM84], Halpern and Moses present an interesting discusssion of knowledge and common knowledge for distributed systems. They argue that while common knowledge is desirable, it is unattainable in certain settings. They suggest a hierarchy of weakened versions of common knowledge and discuss when these can be achieved.

One difficulty with the Halpern and Moses paper is that it is informal and the concepts dealt with are not rigorously defined. Given one interpretation, their theorems are true as claimed, but given another, the opposite occurs, as we show with appropriate counterexamples. Thus, their theorems are not wrong in spirit, but the concepts are subtle and the terms must be more carefully defined.

In the main body of this paper, we give quite general and precise definitions of distributed protocol, knowledge and common knowledge. We also provide precise settings and rigorous proofs for many of the results in [HM84]. We then propose some alternate definitions in which the same results become false.

Our desire is to develop a way to design clear distributed algorithms and write clear proofs about them. We believe we have provided a solid base for future work in this area.

*Research supported in part by the National Science Foundation under Grant Number DCR-8405478 and by the Office of Naval Research under Contract Number N00014-82-K-0154.

†Research supported by an NSF postdoctoral fellowship.

2 Definitions

Definition 2.1 *A* distributed protocol,

$$\mathcal{P} = \langle n, Q, I, \tau \rangle,$$

consists of a number n of participants, a set Q of local states, a set $I \subseteq Q^n$ of initial global states, and a next move relation $\tau \subseteq Q^n \times Q^n$ on global states.

Our definition of protocol is certainly simple and precise. Let us first argue that it is sufficiently general. Anything we would be willing to call a distributed system can be broken up into a finite number of logical entities. Each such entity must be in some total configuration that we are calling its state. Furthermore the states of all the components combined should determine the entire state of the system and thus which global states can be next entered.

It is easy to see for example that our model of distributed system is a generalization of the shared variable model of Lynch and Fischer [LF81]. In that model the components consist of shared variables and processors. Each action involves exactly one processor and one shared variable.

Similarly our model includes synchronous protocols in which every processor sends a message to every other during each round. One way to model this is to specify that the set of possible states is of the form $Q = M^n$, i.e. each processor's total configuration consists of an n-tuple. We can specify that the i^{th} entry of j's state is the value of the message sent from i to j during the previous round. This can be done as follows: for all processors i, j, and for all global states p, q, r, s, if $\langle p, q \rangle$ and $\langle r, s \rangle$ are in τ and if processor i has the same state in p as in r, then the i^{th} component of processor j's state is the same in q as in s.

The sense in which our model could be too general is that we allow any transition relation τ. Of course for certain applications we can make appropriate restrictions. We have already seen that we can restrict our attention to processors which communicate with shared variables, or to synchronous message passing protocols. Similarly, instead of letting each processor's transitions be perfectly general, we can restrict our attention to processors with specified computing power, e.g. finite automaton, polynomial time Turing machine, etc.

For any protocol $\mathcal{P}$, let $R_{\mathcal{P}}$ be the τ-reachable global states of $\mathcal{P}$, that is, the set of all global states we can reach by starting in I and taking any number of τ steps. For $p \in Q^n$ a global state and $1 \leq i \leq n$, we write $(p)_i$ to denote the i^{th} component of p.

For any two states $p, q \in R_{\mathcal{P}}$ and any participant i, we will use the notation $p \overset{i}{\sim} q$ to mean that $(p)_i = (q)_i$, i.e. they are indistinguishable from i's point of view. Obviously each $\overset{i}{\sim}$ is an equivalence relation. For any reachable global state p, define the *i-neighborhood of p* as follows:

$$N_i(p) = \{ q \mid q \overset{i}{\sim} p \}$$

If we are in state p, then all i "knows" is that we are in $N_i(p)$. Therefore, for any sentence α,[1] it is natural to make the following definition of $\mathsf{K}_i\alpha$, which we read as "i knows α":

$$\langle \mathcal{P}, p\rangle \models \mathsf{K}_i\alpha \quad \equiv \quad \forall q \in N_i(p)(\langle \mathcal{P}, q\rangle \models \alpha)$$

Intuitively i knows α just if α is true in all the worlds which are indistinguishable by i from the current world.

It is convenient to picture a protocol $\mathcal{P}$ as a graph with nodes consisting of all the elements of $R_{\mathcal{P}}$. There is a directed edge labelled τ from p to q just if $\langle p, q\rangle \in \tau$. Furthermore there is an undirected edge labelled 'i' between p and q just if $p \overset{i}{\sim} q$.

Let $G \subseteq \{1, \ldots, n\}$ be a group of participants in a protocol. For any $p \in R_{\mathcal{P}}$, define the *G-neighborhood of p* as follows:

$$N_G(p) = \{q \mid (\exists r \geq 0)(\exists i_1, \ldots, i_r \in G)(\exists p_1, \ldots, p_{r-1})[p \overset{i_1}{\sim} p_1 \overset{i_2}{\sim} p_2 \ldots p_{r-1} \overset{i_r}{\sim} q]\}$$

This generalizes our previous definition since $N_i(p) = N_{\{i\}}(p)$.

Analogously to our definition of $\mathsf{K}_i\alpha$, we define $\mathsf{C}_G\alpha$, which we read, "it is common knowledge among the members of G that α":

$$\langle \mathcal{P}, p\rangle \models \mathsf{C}_G\alpha \quad \equiv \quad \forall q \in N_G(p)(\langle \mathcal{P}, q\rangle \models \alpha)$$

We write C for C_G and N for N_G in the special case that G includes all participants.

The next result shows that $\mathsf{C}\alpha$ coincides with the intuitive definition current in the literature. (See for example [HM84].)

Theorem 2.2 *The following two statements are equivalent:*

1. $\langle \mathcal{P}, p\rangle \models \mathsf{C}_G\alpha$.

2. $(\forall r \geq 0)(\forall i_1, \ldots, i_r \in G)(\langle \mathcal{P}, p\rangle \models \mathsf{K}_{i_1}\mathsf{K}_{i_2}\ldots\mathsf{K}_{i_r}\alpha)$.

Proof

($1 \Rightarrow 2$): For any β, we have $\mathsf{C}_G\beta \rightarrow \beta$ since $p \in N_G(p)$. Thus, it suffices to show that for any β, if $\langle \mathcal{P}, p\rangle \models \mathsf{C}_G\beta$, then for all $i \in G$, $\langle \mathcal{P}, p\rangle \models \mathsf{C}_G\mathsf{K}_i\beta$. This is clear because if $q \in N_G(p)$ and $q' \in N_i(q)$ then $q' \in N_G(p)$; hence $\langle \mathcal{P}, q\rangle \models \mathsf{K}_i\beta$. Since $\mathsf{K}_i\beta$ holds for all $q \in N_G(p)$, it is common knowledge in G at p, as desired.

($2 \Rightarrow 1$): Suppose that $\langle \mathcal{P}, p\rangle \not\models \mathsf{C}_G\alpha$. It follows that there is a $q \in N_G(p)$ such that $\langle \mathcal{P}, q\rangle \models \neg\alpha$. Let $i_1, \ldots, i_r \in G$ be such that there exists $p_1, \ldots, p_{r-1}$ with $p \overset{i_1}{\sim} p_1 \overset{i_2}{\sim} p_2 \ldots p_{r-1} \overset{i_r}{\sim} q$. It follows that $\langle \mathcal{P}, p\rangle \models \neg\mathsf{K}_{i_1}\mathsf{K}_{i_2}\ldots\mathsf{K}_{i_r}\alpha$. ∎

[1] We have intentionally left the logical language unspecified from which the sentence α is drawn, for all that we require is that it be possible to interpret α at the pair $\langle \mathcal{P}, p\rangle$.

We conclude this section with two nontrivial examples of protocols, one asynchronous and the other synchronous. These protocols will be frequently referred to in the remainder of the paper.

Let $\mathcal{A} = \langle n^2, Q_{\mathcal{A}}, I_{\mathcal{A}}, \tau_{\mathcal{A}} \rangle$ be an asynchronous message passing protocol defined as follows: The first n participants of $\mathcal{A}$ are the processors $a_1, \ldots, a_n$; the remaining $(n-1)n$ participants are buffers. For a global state p, we abuse our previous notation slightly and write $(p)_{a_i}$ to denote the component corresponding to a_i and $(p)_{b_{i,j}}$ to denote the component corresponding to buffer $b_{i,j}$.

The set of possible local states of a buffer $b_{i,j}$, $i \neq j$, is $M \cup \{\lambda\}$, where M is a set of possible messages and λ is a special symbol denoting the null message. $b_{i,j} = m \in M$ indicates that the single message m was sent by i but not yet delivered to j. $b_{i,j} = \lambda$ indicates that no message is waiting.

The set of possible local states of a processor a_i is $(D \times M^{n-1} \times \mathbf{N})$. State $\langle d, m_1, \ldots, m_{i-1}, m_{i+1}, \ldots, m_n, r \rangle$ indicates that the processor is in internal state d at round r with pending messages $m_1, \ldots, m_{i-1}, m_{i+1}, \ldots, m_n$. If r is even, then the processor is in a 'send' state, waiting to place each $m_j \neq \lambda$ into buffer $b_{i,j}$. If r is odd, then the processor is in a 'receive' state waiting to fetch a message from $b_{j,i}$ for each j such that $m_j = \lambda$.

Thus, the complete set of local states $Q_{\mathcal{A}}$ is $(M \cup \{\lambda\}) \cup (D \times M^{n-1} \times \mathbf{N})$.

The transitions making up $\tau_{\mathcal{A}}$ are of four kinds:

1. $\langle p, q \rangle \in Send_{i,j}$ if
   - $p \overset{c}{\sim} q$ for all $c \notin \{a_i, b_{i,j}\}$;
   - $(p)_{b_{i,j}} = \lambda$;
   - $(q)_{b_{i,j}} = m_j \neq \lambda$;
   - $(p)_{a_i} = \langle d, \ldots, m_{j-1}, m_j, \ldots, 2k \rangle$;
   - $(q)_{a_i} = \langle d, \ldots, m_{j-1}, \lambda, \ldots, 2k \rangle$.
2. $\langle p, q \rangle \in Receive_{i,j}$ if
   - $p \overset{c}{\sim} q$ for all $c \notin \{a_i, b_{j,i}\}$;
   - $(p)_{b_{j,i}} = m_j \neq \lambda$;
   - $(q)_{b_{j,i}} = \lambda$;
   - $(p)_{a_i} = \langle d, \ldots, m_{j-1}, \lambda, \ldots, 2k+1 \rangle$;
   - $(q)_{a_i} = \langle d, \ldots, m_{j-1}, m_j, \ldots, 2k+1 \rangle$.
3. $\langle p, q \rangle \in Stop_i$ if
   - $p \overset{c}{\sim} q$ for all $c \neq a_i$;
   - $(p)_{a_i} = \langle d, \lambda, \ldots, \lambda, 2k \rangle$;
   - $(q)_{a_i} = \langle d, \lambda, \ldots, \lambda, 2k+1 \rangle$.

4. $\langle p, q\rangle \in Start_i$ if

   - $p \overset{c}{\sim} q$ for all $c \neq a_i$;
   - $(p)_{a_i} = \langle d, m_1, \ldots, m_n, 2k+1\rangle$, where $m_j \neq \lambda$ for all $j \neq i$;
   - $(q)_{a_i} = \langle d', m'_1, \ldots, m'_n, 2k+2\rangle$, where $m'_j \neq \lambda$ for all $j \neq i$, and d' and m'_j are functions of $(d, \overline{m}, 2k+1)$.

Now we let $\tau_{\mathcal{A}}$ consist of all the above transitions:

$$\tau_{\mathcal{A}} = \bigcup_{i,j} Send_{i,j} \cup Receive_{i,j} \cup Stop_i \cup Start_i$$

Finally let $I_{\mathcal{A}}$ be some nonempty set of global states in which the state of every processor a_i has the form $(d, m_1, \ldots, m_n, 0)$ with $m_j \neq \lambda$ for all $j \neq i$, and all the buffers are empty.

Our second example of a protocol is a synchronous version of $\mathcal{A}$. Let $\mathcal{B} = \langle n^2, Q_{\mathcal{B}}, \tau_{\mathcal{B}}, I_{\mathcal{B}}\rangle$, where

$$Q_{\mathcal{B}} = \{\lambda\} \cup \left(D \times M^{n-1} \times \{2r \mid r \in \mathbf{N}\}\right)$$

Thus in $\mathcal{B}$ all buffers are empty and the local states of the a_i's are the corresponding states from $\mathcal{A}$ at the beginning of a send phase. Let the transitions $\tau_{\mathcal{B}}$ consist of all pairs $\langle p, q\rangle$ such that there exists a $\tau_{\mathcal{A}}$ path in $\mathcal{A}$ from p to q such that none of the intermediate steps go through global states of $\mathcal{B}$. Finally let $I_{\mathcal{B}} = I_{\mathcal{A}}$.

It is not hard to see that $\mathcal{B}$ is a synchronous version of $\mathcal{A}$ such that in each round all processes send $n-1$ messages and then receive $n-1$ messages.

3 Common Knowledge in Asynchronous Systems

Definition 3.1 *We will call a protocol, $\mathcal{P} = \langle n, Q, I, \tau\rangle$,* totally asynchronous *if for all $\langle p, q\rangle \in \tau$, p and q differ on at most two components.*

The following theorem shows that in a totally asynchronous protocol, no new common knowledge can be acheived.

Theorem 3.2 *Let $\mathcal{P}$ be a totally asynchronous protocol and G a set of at least three participants. Let p be any global state of $\mathcal{P}$ and let p_0 be an intial state from which p is reachable by a sequence of τ steps. Let α be any sentence in a logic for $\mathcal{P}$. If $\langle \mathcal{P}, p\rangle \models \mathsf{C}_G\alpha$ then $\langle \mathcal{P}, p_0\rangle \models \mathsf{C}_G\alpha$.*

Proof We first show that if p is reachable from p_0 by a sequence of τ steps then $N_G(p) = N_G(p_0)$. It suffices to consider the case where $\langle p_0, p\rangle \in \tau$. By the definition of a totally asynchronous protocol, p and p_0 must agree on all but at most two components. Thus, there exists a participant $j \in G$ with the same local state in p_0 as in p, i.e. $p_0 \overset{j}{\sim} p$. It follows that $p \in N_G(p_0)$ and thus $N_G(p) = N_G(p_0)$. The theorem now follows from the definition of common knowledge in G. ∎

As an example, consider the protocol $\mathcal{A}$ discussed at the end of the last section. It is easy to check that $\mathcal{A}$ satisfies the definition of totally asynchronous and thus no new common knowledge can arise in $\mathcal{A}$. By way of contrast, if we look at $\mathcal{A}$'s cousin $\mathcal{B}$, then one observes that all reachable global states in $\mathcal{B}$ have all processors in the same round. Thus, if two reachable global states are $\overset{i}{\sim}$ equivalent for some i, $1 \leq i \leq n$, then they are both in the same round. It follows that if we let $G = \{a_1, \ldots, a_n\}$ be the set of processors—i.e. we don't care what the buffers know—then at any round r, 'C_G(we're at round r)' holds, i.e. it is common knowledge in G that all processors are at round r.[2]

It would seem at first glance that the difficulty in achieving common knowledge has to do with the problem of reaching an arbitrary depth of K's with only finitely many messages. We conclude this section with a look at finite state protocols where common knowledge is equivalent to a finite stack of K's.

Theorem 3.3 *Let $\mathcal{P} = \langle n, Q, I, \tau\rangle$ be a finite state protocol, i.e. $|Q| < \infty$. For each i, let*

$$Q_i = \{(q)_i \mid q \in R_{\mathcal{P}}\}$$

Thus each processor is a $|Q_i|$ state automaton. Let $r = \min\{|Q_i| \mid 1 \leq i \leq n\}$. Let p be any global state and let α be any formula. Then the following are equivalent:

1. *$\langle \mathcal{P}, p\rangle \models \mathsf{C}\alpha$.*
2. *For all $i_1, i_2, \ldots, i_{2r-1}$, $(\langle \mathcal{P}, p\rangle \models \mathsf{K}_{i_1} \ldots \mathsf{K}_{i_{2r-1}}\alpha)$.*

Proof

$(1 \Rightarrow 2)$: By definition of C.

$(2 \Rightarrow 1)$: Suppose that $\langle \mathcal{P}, p\rangle \not\models \mathsf{C}\alpha$. Then there must exist $q \in N(p)$ such that $\langle \mathcal{P}, q\rangle \models \neg\alpha$. Consider a minimum length $\sim$ chain from p to q:

$$p = p_0 \overset{i_1}{\sim} p_1 \overset{i_2}{\sim} p_2 \ldots p_{s-1} \overset{i_s}{\sim} p_s = q$$

Note that no nonconsecutive pair p_j, p_k can agree on some component because if they did the chain could be shortened. It follows that in any given component each state appears at most twice. Therefore $s \leq 2r - 1$. It follows that

$$\langle \mathcal{P}, p\rangle \models \neg\mathsf{K}_{i_1}\mathsf{K}_{i_2} \ldots \mathsf{K}_{i_{2r-1}}\alpha$$

[2] This assumes of course that our logical language is powerful enough to express the property 'we're at round r'.

∎

Example 3.4 *Consider the protocol* $\mathcal{P}_r = \langle 2, \{1,\ldots,r+1\}, \{\langle 1,1\rangle\}, \tau_r\rangle$ *where,*

$$\tau_r = \{(\langle i,i\rangle, \langle i+1,i\rangle) \mid 1 \le i \le r\} \cup \{(\langle i+1,i\rangle, \langle i+1,i+1\rangle) \mid 1 \le i < r\}.$$

This protocol has the unique computation chain:

$$\langle 1,1\rangle,\ \langle 2,1\rangle,\ \langle 2,2\rangle,\ \langle 3,2\rangle,\ldots,\langle r,r-1\rangle,\ \langle r,r\rangle,\ \langle r+1,r\rangle.$$

Furthermore, for all global states p *and* q *we have* $N(p) = N(q)$. *Thus, for any* α,

$$\langle \mathcal{P}_r, p\rangle \models \mathsf{C}\alpha \Leftrightarrow \text{ for all } q \in R_{\mathcal{P}_r}, \langle \mathcal{P}_r, q\rangle \models \alpha.$$

Let α *say that processor 1 is not in state 1. Then* $\langle \mathcal{P}_r, \langle 1,1\rangle\rangle \not\models \alpha$, *so* $\langle \mathcal{P}_r, \langle r+1,r\rangle\rangle \not\models \mathsf{C}\alpha$. *On the other hand, it is easily seen that*

$$\langle \mathcal{P}_r, \langle r+1,r\rangle\rangle \models \mathsf{K}_1 \underbrace{\mathsf{K}_2\mathsf{K}_1\mathsf{K}_2\mathsf{K}_1\ldots\mathsf{K}_2\mathsf{K}_1}_{2r-2}\alpha$$

It follows that for all $i_1, i_2, \ldots, i_{2r-2} \in \{1,2\}$,

$$\langle \mathcal{P}_r, \langle r+1,r\rangle\rangle \models \mathsf{K}_{i_1}\ldots\mathsf{K}_{i_{2r-2}}\alpha,$$

showing that the bound in Theorem 3.3 cannot be improved.

4 Alternate Definitions of Knowledge

According to Halpern and Moses, "If Cp is to be attained, all processors must start supporting it simultaneously."[3] Unfortunately the notion of two distant events occuring simultaneously has no meaning in modern physics. Do Halpern and Moses plus Einstein imply that no real distributed system ever achieves new common knowledge? A corollary would be that no real, synchronous distributed system can exist.

A look at our example protocols $\mathcal{A}$ and $\mathcal{B}$ reveals that they are realistic. Recall that new common knowledge among the n processors is attainable in $\mathcal{B}$ but not in $\mathcal{A}$. This is all the more confusing because in a very strong sense $\mathcal{A}$ and $\mathcal{B}$ are isomorphic protocols (cf. [CM85]).[4]

The difference between protocols $\mathcal{A}$ and $\mathcal{B}$ concerns the granularity at which processors in the two protocols may introspect. In $\mathcal{B}$, processors are only allowed

[3][HM84], Lemma 2.

[4]We will call a pair of protocols such as $\mathcal{A}$ and $\mathcal{B}$, all of whose interactions are accomplished by a series of messages, ***isomorphic*** if the set of messages sequences they generate is identical up to permutations which do not switch the order of a send and a receive by the same participant, nor the order of a send and its corresponding receive.

to think about what they know at the start of each write phase. When two isomorphic structures differ on some property, we become very suspicious about whether or not that property is well defined. In the present case we must reexamine our definitions of knowledge and common knowledge.

Let $\mathcal{P}$ be any protocol and let $S \subseteq R_{\mathcal{P}}$ be any subset of reachable global states. For each i, let $\overset{i}{\sim}_S$ be the restriction of $\overset{i}{\sim}$ to $S \times S$. We can now generalize our previous definition of neighborhood. Let $G \subseteq \{1,\ldots,n\}$ be a group of the participants in a protocol. For any $p \in S$, define the *G-neighborhood of p with respect to S* as follows:

$$\begin{aligned} N_G^S(p) \quad = \quad & \{q \mid (\exists r \geq 0)(\exists i_1,\ldots,i_r \in G)(\exists p_1,\ldots,p_{r-1} \in S) \\ & \quad [p \overset{i_1}{\sim}_S p_1 \overset{i_2}{\sim}_S p_2 \ldots p_{r-1} \overset{i_r}{\sim}_S q]\} \end{aligned}$$

Intuitively, i knows only about the global states in S. We thus define $\mathsf{K}_i^S \alpha$ as follows: for $p \in S$,

$$\langle \mathcal{P}, p \rangle \models \mathsf{K}_i^S \alpha \quad \equiv \quad \forall q \in N_i^S(p)(\langle \mathcal{P}, q \rangle \models \alpha).$$

Similarly, we can define common knowledge in G with respect to S:

$$\langle \mathcal{P}, p \rangle \models \mathsf{C}_G^S \alpha \quad \equiv \quad \forall q \in N_G^S(p)(\langle \mathcal{P}, q \rangle \models \alpha).$$

It is easy to see that the following generalization of Theorem 2.2 holds:

Theorem 4.1 *Let $S \subseteq R_{\mathcal{P}}$ and let $G \subseteq \{1,\ldots,n\}$. For $p \in S$ the following two statements are equivalent:*

1. $\langle \mathcal{P}, p \rangle \models \mathsf{C}_G^S \alpha$.
2. $(\forall r \geq 0)(\forall i_1,\ldots,i_r \in G)(\langle \mathcal{P}, p \rangle \models \mathsf{K}_{i_1}^S \mathsf{K}_{i_2}^S \ldots \mathsf{K}_{i_r}^S \alpha)$.

The following theorem shows that for any protocol $\mathcal{P}$ and any nonempty $S \subseteq R_{\mathcal{P}}$, the operators K^S and C^S satisfy the standard S5 axioms for knowledge operators. It follows that if we consider the protocol $\mathcal{A}$ with $S = R_{\mathcal{B}}$, then we get a quite reasonable definition of knowledge and common knowledge for which the asynchronous protocol $\mathcal{A}$ does attain new common knowledge. This contradicts Theorem 3 of [HM84]. More importantly, these observations show the definitions of knowledge and common knowledge needed to make useful progress in the understanding of distributed protocols are much more subtle than one might have at first thought.

Theorem 4.2 *For any protocol $\mathcal{P}$, any nonempty $S \subseteq R_{\mathcal{P}}$, and $G \subseteq \{1,\ldots,n\}$, the operators K_i^S and C_G^S satisfy the standard S5 axioms for modal operators.*

Proof This is immediate from the fact that each $\overset{i}{\sim}_S$ is an equivalence relation. ∎

5 Conclusions

We have given precise formulations of distributed protocols. For any subset S of the reachable states, we have given a precise definition of knowledge and common knowledge with respect to S. We have presented theorems outlining some cases where new common knowledge can be attained and some cases where it cannot. Most strikingly, we have shown that in some situations two plausible choices for S can give completely different results.

One can now ask the question, "For which sets of protocols is there a 'best' choice for S?" and thus a 'best' definition for knowledge and common knowledge. We suspect that in at least certain situations there may be such a best S, and that in this case knowledge and common knowledge with respect to S may be valuable tools.

Many arguments in distributed systems are first formulated at the intuitive level of what certain processors 'know' at certain points in the computation. With precise definitions for these concepts, it may be easier to formulate clear and correct proofs. We believe that considerable work is needed in order to develop logical tools and demonstrate their usefulness on problems of interest in distributed systems.

6 References

References

[CM85] K. Mani Chandy and Jayadev Misra, "How Processes Learn," *Fourth ACM Symp. on Principles of Distributed Computing* (1985), 204–214.

[HM84] J. Y. Halpern and Y. Moses, "Knowledge and Common Knowledge in a Distributed Environment," *Third ACM Symp. on Principles of Distributed Computing* (1984), 50–61.

[LF81] Nancy A. Lynch and Michael J. Fischer, "On Describing the Behavior and Implementation of Distributed Systems," *Theoretical Comp. Sci. 13* (1981), 17–43.

DISTRIBUTION LIST

Office of Naval Research Contract N00014-82-K-0154

Michael J. Fischer, Principal Investigator

Defense Technical Information Center
Building 5, Cameron Station
Alexandria, VA 22314
(12 copies)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Dr. R.B. Grafton, Scientific Officer (1 copy)

Information Systems Program (437) (2 copies)

Code 200 (1 copy)
Code 455 (1 copy)
Code 458 (1 copy)

Office of Naval Research
Branch Office, Pasadena
1030 East Green Street
Pasadena, CA 91106
(1 copy)

Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375
(1 copy)

Dr. A.L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380
(1 copy)

Naval Ocean Systems Center
Advanced Software Technology Division
Code 5200
San Diego, CA 92152
(1 copy)

Mr. E.R. Gleissner
Naval Ship Research and Development Center
Computation and Mathematics Department
Bethesda, MD 20084
(1 copy)

Captain Grace M. Hopper
Naval Data Automation Command
Washington Navy Yard
Building 166
Washington, D.C. 20374
(1 copy)

Defense Advance Research Projects Agency
ATTN: Program Management/MIS
1400 Wilson Boulevard
Arlington, VA 22209
(3 copies)