2002

TR-2002006: States of Knowledge

Rohit Parikh

States of Knowledge

Rohit Parikh
City University of New York[1]
Preliminary report (*WOLLIC-2002*)

> A travelling salesman found himself spending the night at home with his wife when one of his trips was accidentally cancelled. The two of them were sound asleep, when in the middle of the night there was a loud knock at the front door. The wife woke up with a start and cried out, 'Oh my God! It's my husband!' Whereupon the husband leapt out of bed, ran across the room and jumped out the window.
>
> Schank and Abelson, 1977, p. 59.

Wimmer and Perner begin their paper [WM] on Beliefs about beliefs with this story from Schank and Abelson which may seem amusing to some and disturbing to others. But the point of the story seems to be that husband and wife each have their own scenario and neither corresponds to the actuality.

Wimmer and Perner themselves are concerned primarily with the perception by children of other people's *mindsets*. The following quote from [WM] is a story (in Austrian English) about Maxi which they told a group of children:

> Mother returns from her shopping trip. She bought chocolate for a cake. Maxi may help her put away the things. He asks her 'Where should I put the chocolate?' In the blue cupboard, says the mother.

Later, with Maxi gone out to play, the mother transfers the chocolate from the blue cupboard to the green cupboard. Maxi then comes back from the playground, hungry, and he wants to get some chocolate.

In Wimmer and Perner's experiment, little children who were told the Maxi story were then asked the BELIEF question, "Where will Maxi look for the chocolate?"

Children at the age of 3 or less invariably got the answer wrong and assumed that Maxi would look for the chocolate in the *green* cupboard where *they* knew it was. Even children aged 4-5 had only a one third chance of correctly answering this question or an analgous question involving Maxi and his brother (who also wants the chocolate and whom Maxi wants to deceive). Children aged 6 or more were by contrast quite successful in realizing that Maxi would think the chocolate would be in the *blue* cupboard – where he had put

[1]Email: rparikh@gc.cuny.edu Research supported by grants from the NSF and the CUNY-FRAP program

it and that if he wanted to deceive his brother, he should lead his brother towards the green cupboard.

Thus it seems that representation of other people's mindset comes fairly late in childhood, well after they have learned to deal with notions of belief and belief based action for themselves and for others who share their own view of reality. In [CS] Chris Steinsvold investigates modal logics which are intended to represent the states of mind of young children. See also [SP].

Older children are not much better. In an experiment in my daughter's 7th grade class, I found that they were unable to deal with the muddy children puzzle beyond the first one or two levels.

In this, by now well known puzzle, a number of children are playing in the mud and some of them get their foreheads dirty. At this the father comes on the scene and announces, "at least one of you has got her forehead dirty".

Scenario 1: Suppose there is only one child, say Amy, who is dirty. Then she will realize that her own forehead must be dirty since she can see that the others are clean.

Scenario 2: Suppose now that there are two dirty children, Sarah and Amy, who are asked in turn, "Do you know if your forehead is dirty?" Now when Sarah is asked, she can see Amy's dirty forehead and she replies, "I don't know". However, when Amy is asked, she is able to reason, "If *my* forehead were clean, Sarah would have known that hers must be dirty since all the others are clean. But Sarah did not know. So my forehead must be dirty."

This reasoning on Amy's part requires a representation by Amy of Sarah's state of mind, and clearly Amy must be at least six for this to work. However, Sarah herself must have some reasoning ability and Amy must know that she has such abilities. It is not enough for Amy to know Sarah's view of reality, she must also represent Sarah's logical abilities in her own mind.

As the number of dirty children goes up, there is a need for higher and higher levels of "I know that he knows that she knows that...". Common knowledge is at the end of this road and has been offered as the explanation of co-ordinated behaviour ([Lew, HM, CM]). For instance Halpern and Moses in [HM] show that the co-ordinated attack problem requires common knowledge between the two generals, and that given the means of communication they have, such common knowledge is impossible to attain. Clark and Marshall indicate similar difficulties with the referent of "the movie playing at the Roxy today".

While it is true that co-ordinated actions and, supposedly, common knowledge do happen, it may also be relevant to consider other levels of knowledge, short of the infinite, common knowledge, level.[2] Such levels also arise in certainly pragmatic situations, e.g. with email

[2]The following, possibly apocryphal story about the mathematician Norbert Wiener, who was well known for his absent mindedness, illustrates something even more subtle. At one time the Wieners were

or snailmail or messages left on telephones as voice mail. Thus the purpose of this paper is to study levels other than common knowledge.

In typical co-operative situations even if a certain level of knowledge is needed, a higher level would also do. If Bob wants Jill to pick up the children at 4 PM, it is enough for him to know that she knows. Thus if he sends her email at 2 PM and knows that she always reads hers at 3 PM, he can be satisfied. In such a situation Bob knows that Ann will know about the children in time, or symbolically $K_b(K_a(C))$ and he may feel this is enough. However, if he telephones her at 3 PM instead, this will create common knowledge of C, much *more* than is needed. But no harm done, since in this context, Ann and Bob have the same goals. Halpern and Zuck also state a knowledge level requirement for the sequence transmission problem, which suffices as a minimum, but since the parties are co-ordinating, a higher level does not harm.

But in other contexts one may wish for just a particular level of knowledge, no lower, and no *higher*. Suppose for instance that Bob wants Ann to know about a seminar talk he is giving, just in case she wants to come, but he does not want her to feel pressured to come – she should come only out of interest and not from politeness. In that case he will want to arrange that $K_b(K_a(S))$, (he himself knows that she knows about the seminar) but not $K_a(K_b(K_a(S)))$ (Ann knows that Bob knows that Ann knows about the seminar), for in the latter case she *would* feel pressured. Instead of telling her about his talk, which would create common knowledge, he may arrange for some other method, perhaps for a student to tell her, but without saying that it is a message from Bob.

Suppose a pedestrian is crossing the street and sees a car approaching him. It happens in many cities, Boston, Naples, etc., that the pedestrian will pretend not to notice the car, thereby preventing $K_dK_p(C)$ with C representing the car, d being the driver and p the pedestrian. If the driver knew that the pedestrian knew, he might drive aggressively and try to bully the pedestrian into running or withdrawing. But if he does not know that the pedestrian knows, he will be more cautious.

While the social questions are fascinating and are addressed elsewhere (Cf. [Pa3]), in this paper we shall concentrate on the technical aspects of knowledge, where it is assumed that everyone involved is logically perfect. One can still ask, what are the various levels of knowledge which can arise under various circumstances of communication?

---

moving and in the morning as he was going to work, Mrs. Wiener said to him, "Now don't come home to this address in the evening." And she gave him a piece of paper with the new address. However, in the evening Wiener found himself standing in front of the old address and not knowing what to do – he had in the meanwhile lost the slip of paper with the new address. He went to a little girl standing by and said, "Little girl, do you know where the Wieners have moved to?" The little girl replied, "Daddy, Mom knew what would happen so she sent me to fetch you." The moral of the story, for *us*, is that common knowledge works only if the memory of all parties involved is reliable.

1 Model of a distributed system

Note: most of the results which follow are joint with Paul Krasucki, except where indicated, and full proofs are available in [PK].

We assume that there are a finite number of processes, $1, ..., n$, which compute and communicate with each other either by asynchronous messages or by broadcasts. Our network is assumed to be fully connected[3] (there is a channel from every process to every other process).

Asynchronous communication consists of two phases: send and receive. All messages sent are ultimately delivered (and they are delivered in the order in which they were sent) but the delay (transmission time) may be arbitrarily long.

Broadcasts are fully reliable, synchronous communications[4] where all processes involved simulteneously receive the message sent by one of them.

Now we formally specify our class of models. Let $N = \{1, ..., n\}$ be the set of all processors. Every processor i has infinitely many possible initial states v. Every initial state is a string of 0's and 1's ($v \in \{0,1\}^*$). The set of initial states for i we denote by V_i. The set of global initial states is $\mathcal{V} = \prod_{i=1}^{n} V_i$.

From now on we will use lower case letters to denote everything pertaining to a single process. Capitals will be used where all the processes are involved (e.g. v_i is an initial state of a processor i, while V is an initial configuration of the whole system: $V = (v_1, ..., v_n)$).

Events: E_i denotes the set of all events in which processor i can participate (events *local* to i). There are the following types of events (or actions):

1. L_i: Local computation steps.
2. $s(i, j, m)$: Sending a message m to a processor j, $j \in N$.
3. $r(j, i, m)$: Receiving a message m from a processor j, $j \in N$.
4. $bc(i, U, m)$: Sending a broadcast m to a group of processors U, $i \in U \subseteq N$. The same event is receiving a broadcast m by a group of processes U.

$$E_i \;=\; L_i \cup \{s(i,j,m) | m \in M,\ j \in N\} \cup \{r(j,i,m) | m \in M,\ j \in N\}$$

[3] If the network is not fully connected then some levels of knowledge may be impossible to realize due to the lack of communication capabilities, e.g. if a processor is isolated (cannot communicate with anyone)then the other processes cannot learn anything from that process. Interesting questions arise in case of a directed network where every process may communicate with every other process but some communications are necessarily indirect (go through other processes). We will not analyze this case here.

[4] The two kinds of communications can be looked at as two kinds of communication media e.g. mailing system (asynchronous) and telephone lines (synchronous). Since we allow for synchronous communication between more than two processes at a time, our telephone system must have "conference call" capability.

$$\cup \{bc(j,U,m) | m \in M,\ i,j \in U \subseteq N\} \cup \{bc(i,U,m) | m \in M,\ i \in U \subseteq N\}$$

We define the set of *global events* $\mathbf{G}$ in our system. $\mathbf{G} \subseteq \prod_{i=1}^{n}(E_i \cup \{null\})$ (a cartesian product) s.t. if $(e_1, ..., e_i, ..., e_n) \in \mathbf{G}$ for some i and $e_i = bc(j,U,m)$ then for all $i' \in U$, $e_{i'} = bc(j,U,m)$. If $e_i = null$ for some i, it means that there is no local event at i at this point. Note that *null* is *not* local to any process. We use the notation $(G)_i$ to denote the ith coordinate of G, so $(e_1, ..., e_i, ..., e_n)_i = e_i$.

Histories: A history (a run) is an input value followed by a sequence of events. Let's call the set of all possible histories of the system – a protocol $\mathbf{P}$. So $\mathbf{P} \subseteq V; \mathbf{G}^*$. Protocols are always closed under taking an initial segment of a history: $H \in \mathbf{P}$ implies that every H' which is an initial segment of H is in $\mathbf{P}$.

We will require that for every receive in every history in every protocol there is exactly one corresponding send and it occurs before receive (this condition we will call *time-consistency*).

We say that two histories H and H' are *compatible* iff they start with the same input values.

We can define the *concatenation* of compatible histories:
If $H_1 = V; G_1; \ldots; G_k$, and $H_2 = V; G'_1; \ldots; G'_l$, then H is the concatenation of H_1 and H_2 iff $H = V; G_1; \ldots; G_k; G'_1; \ldots; G'_l$,

Local histories are the projections of global histories onto the sets of local events of the processors. They are "time-forgetting".

We assume that a global event – the ticking of the clock – takes place even if no local events take place at a particular moment. Given i, and the global history H, the local history h_i consisting of the events seen by i, is uniquely defined and we let Φ_i be the map which takes us from H to h_i.

The local history is everything the processor sees, so all the global histories which correspond to the same local history h_i look the same to the processor i. Note that the length of $\Phi_i(H)$ is less than or equal to the length of H. In fact $length(\Phi_i(H)) = length(H)$ iff there are no *null* events on i in H.

For every i we can define an equivalence relation on the set of global histories:

$$H \approx_i H' \text{ iff } \Phi_i(H) = \Phi_i(H')$$

This relation is extended to groups U by letting $H \approx_U H'$ iff there exists a chain $H = H_1, H_2, ..., H_m = H'$ and for all $i < n$, there is a $j \in U$ such that $H_i \approx_j H_{i+1}$.

We use capital letters to denote global histories, events etc, lower case letters do denote local histories, events etc.

Closure Conditions for the Protocol: We impose some additional conditions on the protocol $\mathbf{P}$. We want to ensure that the initial state of i (v_i) cannot be *known* to any other

process j at any run of the system, unless j *learns* about v_i from some communication. We want to exclude the possibility that something is common knowledge "accidentally". To achieve that we will make sure that all the initial states are possible. Moreover, if v_i is the initial state of i, all other strings v_i' will remain possible for j as initial states of i, unless j gets some message from i to the contrary (directly or via some other processors).

1) All vectors of input values are possible: $\forall V$ s.t. $V = (v_1, \ldots, v_n)$ where every v_i is a sequence of 0's and 1's there is some $H \in \mathbf{P}$ s.t. for some H', $H = V; H'$.

2) No sequence of local events on some group of processes can influence possible actions of some other group of processes unless there are some communications (of course assuming that both groups are disjoint).

For that we need some closure conditions on the set of all protocols. The first condition we use is due to [CM] (it is the first of their *principles of computation extension*).

We need one definition:
Let $G = (e_1, \ldots e_n)$, G is *on* U if $U = \{i | (G)_i \neq null\}$ (so U is the set of processes which have some local events in G).

Closure conditions:

(i) **Extension Rule:**
Let $\forall i \in U$, $H \approx_i H'$, G is on U, none of $(G)_i$ is receive $r(j, i, m)$ for any j not in U, then

$$(H' \in \mathbf{P},\ H; G \in \mathbf{P}) \quad \Rightarrow \quad H'; G \in \mathbf{P}$$

The extension rule guarantees that if we have a protocol $\mathbf{P}$, some history H in $\mathbf{P}$ and some action of a group of processes U is possible in H, then the same action must be possible in every history H' which looks the same to all processes in U unless it violates time–consistency. In order to explain why e_i cannot be a receive from a processor outside of U let us examine an example:

Let $N = \{1, 2, 3\}$, $U = \{1, 2\}$.
$H = (null, null, s(3, 1, m))$, $H' = (null, null, null)$. Clearly $H \approx_1 H'$ and $H \approx_2 H'$. If we take $G = (r(3, 1, m), null, null)$ s.t. $H; G \in \mathbf{P}$ then requiring $H'; G$ to be in $\mathbf{P}$ would violate time–consistency.

The following conditions ensure that no process can get any additional information about the other processes by observing its own local events (no hidden synchronization). These conditions are necessary because (unlike [CM]) we allow local events at different sites at the same instant of time. Condition (ii) says that if some local events have occurred in parallel, and the sets of participating processes were disjoint, they could have occurred in sequence. We'll call it the splitting rule.

(ii) **Splitting Rule:**
$G = (e_1, \ldots e_n)$, $G \notin V$, G is on U. Given U_1, U_2 s.t. $U_1 \cup U_2 = U$ and U_1, U_2 disjoint,

then we can "split" any G into G_1 and G_2:

$$(H; G \in \mathbf{P}) \quad \Rightarrow \quad H; G_1; G_2 \in \mathbf{P}$$

where $(G)_i = (G_1)_i$ for $i \in U_1$, $(G)_i = (G_2)_i$ for $i \in U_2$, $(G_1)_j = null = (G_2)_k$ for $j \notin U_1$, $k \notin U_2$ provided that we don't split any broadcasts: $(G)_i = bc(i, V, m) \rightarrow V \subseteq U_1 \vee V \subseteq U_2$.

Condition (*iii*) says that if some local events have occurred in sequence, the sets of participating processes were disjoint, and there was no send receive pair in them, they could have occurred in parallel.

(iii) **Joining Rule:**
Given U_1, U_2 s.t. $U_1 \cup U_2 = U$ and U_1, U_2 disjoint, Let G_1 be on U_1, G_2 on U_2, and if $(G_1)_i = s(i, j, m)$ then $(G_2)_j \neq r(i, j, m)$.

$$(H; G_1; G_2 \in \mathbf{P} \quad \Rightarrow \quad H; G \in \mathbf{P})$$

where $(G)_i = (G_1)_i$ for $i \in U_1$, $(G)_i = (G_2)_i$ for $i \in U_2$.

Systems: We consider three kinds of systems. *Asynchronous* systems are the systems as described above but *without broadcasts*. So in asynchronous systems the only communications are via send and receive. *Synchronous* systems are the systems in which all the communications are done using broadcasts where we don't have the events *send* and *receive*. Finally, we use the name *mixed* communications systems for the systems with both kinds of communications available.

1.1 Language and Semantics

Let L_0 be a language which describes properties of the global histories in a protocol $\mathbf{P}$. So for every sentence A in L_0, and for every history $H \in \mathbf{P}$, A is either true or false in H.

We want to make sure that in every history initially every processor has some "private" information not known to any other processor. To accomplish that we assume that we have in our language a countable set of propositions $L_1 = \{Q_{i,j}\}_{i,j \in N}$. $Q_{i,j}$ is the proposition that the jth input value of i is 1. All $Q_{i,j}$ are independent. Private information of i in H are $P_{i,j}$ which are $Q_{i,j}$ or its negation depending on whether $Q_{i,j}$ is true in H or not. Note that the private information is not *a truth value* of any formula, but *which* formula we're looking at.

L is the closure of L_0 under truth functional connectives. L can be extended to a larger language L_C which is the closure of L under common knowledge operators C_U (for $U \subseteq N$) and the usual truth functional connectives. $C_U(A)$ means that there is common knowledge of A among processes from U.

The knowledge of a single process corresponds to $C_{\{i\}}$. We will then use the notation[5] K_i for $C_{\{i\}}$. When we restrict ourselves to a subset of L_C in which all common knowledge operators are in fact the knowledge operators (the sets U in C_U are always singletons) then we use the notation L_K.

The class of all models we consider is the class of all protocols $\mathbf{P}$ as described in the previous section. Fix $\mathbf{P}$. Now we define the notion $H \models A$ for A in L^+ by recursion on the complexity of A.

0) If A is from L_0 then the semantics is given.
1) If A is $Q_{i,j}$ then A is true in H if the jth bit of an input of processor i in H is 1:

$$H \models A \quad \text{iff} \quad H = (v_1, \ldots, v_n); H', \ (v_i)_j = 1$$

2) If A is $\neg A'$ then

$$H \models A \quad \text{iff} \quad H \not\models A'$$

If A is $B \vee C$ then

$$H \models A \quad \text{iff} \quad (\ H \models B \ \text{ or } \ H \models C\ )$$

3) If A is of the form $K_i(B)$ then

$$H \models K_i A \quad \text{iff} \quad \forall H' \in \mathbf{P} \quad H \approx_i H' \quad \rightarrow \quad H' \models A$$

4) If A is of the form $C_U(B)$, then

$$H \models A \ \text{ iff } \ \text{for all } \ H', H' \approx_U H, H' \models B$$

Also if U is empty, then $C_U A$ iff A.

Theorem 1 : Let Σ_C be the alphabet whose symbols are $\{C_U\}_{U \subseteq N}$ For all x, y in Σ_C^*, and all formulae A, for all H, $V \subseteq U \subseteq N$, $H \models xC_UC_VyA$ iff $H \models xC_VC_UyA$ iff $H \models xC_UyA$.

Corollary 1 : Let Σ_K be the alphabet whose symbols are $\{K_1, \ldots, K_n\}$ For all a in Σ_K, and for all x, y, in Σ_K^*, and all formulae A,

$$\vdash xayA \leftrightarrow xaayA$$

and hence for all H, $H \models xayA$ iff $H \models xaayA$. I.e. repeated occurrences of a are without effect and if $xay \in T_K(A, H)$ then $\forall n \ xa^n y \in T_K(A, H)$.

Definition : Given a formula A and a history H, the *level* of A at H, $L(A, H)$ is the set of x in Σ_C^* such that $H \models xA$, and x contains no substrings C_UC_V, C_VC_U for any $V \subseteq U \subseteq N$.

If H is clear from the context, or not important, then we shall drop it as a parameter. If we restrict ourselves to the K_i operators, we denote the level of A in H by $L_K(A, H)$.

[5]Fact that $C_{\{i\}} = K_i$ was noticed earlier, compare e.g. [FI]. It is important that we assume that L_K and L_C are S5 (we need at least S4).

2 Embeddability

Now we will try to characterize levels of knowledge. First we need to introduce the embeddability ordering on strings which turns out to be important here.

Definition : Given two strings x and y, we say that x is *embeddable* in y ($x \leq y$), if all the symbols of x occur in y, in the same order, but not necessarily consecutively. Formally:
1) $x \leq x$, $\epsilon \leq x$ for all x
2) $x \leq y$ if there exist x', x'', y', y'', $(y', y'' \neq \epsilon)$, such that $x = x'x''$, $y = y'y''$, and $x' \leq y'$, $x'' \leq y''$.
and $\leq$ is the smallest relation satisfying (1) and (2).

Thus the string *aba* is embeddable in itself, in *aaba* and in *abca*, but not in *aabb*.

Properties of the embeddability relation $\leq$

Fact 1: Embeddability is a *well partial order*, i.e. it is not only well founded, but every linear order that extends it is a well order (equivalent condition: it is well founded and every set of mutually incomparable elements is finite).

Fact 2: Embeddability can be tested in linear time, e.g by a nondeterministic finite automaton with two input tapes.

For a proof of fact 1 see [H]. Fact 2 is straightforward.

We also need a stronger relation defined on Σ_C^*, which we call *C–embeddability*.

Definition : Given two strings x and y, we say that x is *C–embeddable* in y ($x \preceq y$), if
1) If $V \subseteq U$ then $C_V \preceq C_U$
2) $x \preceq y$ if there exist x', x'', y', y'', $(y', y'' \neq \epsilon)$, such that $x = x'x''$, $y = y'y''$, and $x' \preceq y'$, $x'' \preceq y''$.
and $\preceq$ is the smallest relation satisfying (1) and (2).

Fact 3: For any $x, y \in \Sigma^*$, $x \leq y$ iff $x \preceq y$.

Fact 4: C-embeddability is a well partial order.

Fact 3 is easy. It is also easy to check that C-embeddability is a partial order. It is well founded, because regular embeddability is well founded and for given $x \in \Sigma_C^*$ there are only finitely many $y \in \Sigma_C^*$ s.t. $|x| = |y|$ and $y \preceq x$.
There are only finitely many incomparable elements in Σ_C^* with respect to $\leq$, and there are more incomparable elements with respect to $\leq$ than with respect to $\preceq$, so $\preceq$ is a well partial order. □

If $\leq$ is a partial order on S, we can define a notion of a *downward closed* subset of S:

Definition : $R \subseteq S$ is *downward closed* iff $x \in R$ implies $\forall y \leq x, \quad y \in R$.

We will look at downward closed sets with respect to embeddability and C-embeddability.

Theorem 2 : Let Σ_C be the alphabet whose symbols are $\{C_U\}_{U\subseteq N}$. Then for all strings x, y Σ_C^*, if $x \preceq y$ then for all histories H, if $H \models yA$ then $H \models xA$.

3 The Main Results

Corollary 1 : Every level of knowledge is a downward closed set with respect to $\preceq$. □

Theorem 3 : There are only countably many levels of knowledge and in fact all of them are regular subsets of Σ^* (where Σ is either Σ_K or Σ_C). □

Fact 5: Eric Pacuit of the CUNY Graduate center and ourselves have shown that in contrast with *knowledge* there are *uncountably* many possible levels of rational *belief*. This is curious as truth is the only condition which (formally) separates knowledge from rational belief. These results will appear elsewhere.

Corollary : The membership problem for a level of knowledge can be solved in linear time.

Theorem 4 : If L is a non-empty finite subset of Σ_K^*, then L is downward closed iff for some k,

$$L = \bigcup_{i=1}^{k} dc(\{x_i\})$$

where $x_i \in \Sigma_K^*$. This theorem reiterates the fact that the finite levels are characterized by their maximal elements ($x_1, ..., x_k$ are maximal).

Definition: A formula A is *persistent* if whenever $H \models A$ and H' extends H, then $H' \models A$.

Theorem 5 : If A is persistent then so is $K_i(A)$ for any i.

Theorem 6: Every formula A which is a boolean combination of P_i's is persistent. □

Theorem 7: Every formula of the form xA where A is a boolean combination of P_i's, and x is a string of knowledge operators is persistent. □

Theorem 8 [Chandy, Misra]: If communication is purely asynchronous, and for some histories H, H', s.t. H is an initial segment of H':

$$H' \models K_1K_2...K_nA \text{ and } H \not\models K_nA$$

then in $H' - H$ there must be a sequence of messages: $m_{n-1}, m_{n-2}, \ldots, m_1$ s.t. m_{n-1} is sent by n and reaches $n-1$ (maybe via some other processes),...,m_1 is sent by 2 and (maybe indirectly) reaches 1 (messages may be different but they all must imply A).

Moreover if A doesn't depend on any local event of n (its truth value depends on some event $e \notin E_n$) then there must be some event of the form $r(i, n, m)$ occurring after H but before $s(n, n-1, m_{n-1})$.

Theorem 9 : Every finite downward closed set is the set $L(A, H)$ for an appropriate A and H in some asynchronous protocol.

Theorem 10 : Every downward closed set L of strings without repetitions is $L(A, H)$ for suitable A and H in a synchronous system with at least 3 processors.

Theorem 11: In a two processor system with only synchronous communication available, no finite level containing strings of length ≥ 2 can be achieved for any formula A.

Theorem 12: In system with k-casts, i.e. with broadcasts involving at most k processors, it is impossible to achieve common knowledge of any new fact in a group of size $> k$.

References:

[Bar] J. Barwise, Three Views of Common Knowledge, in *TARK-2*, Ed. M. Vardi, Morgan Kaufmann 1988, pp. 369-380.

[CM] Chandy M. and Misra J., "How Processes Learn", *Proceedings of 4th ACM Conference on Principles of Distributed Computing* (1985) pp 204-214.

[CM2] H. H. Clark and C. R. Marshall, Definite Reference and Mutual Knowledge, in *Elements of Discourse Understanding*, Ed. Joshi, Webber and Sag, Cambridge U. Press, 1981.

[FI] M. Fischer and N. Immerman, "Foundations of Knowledge for Distributed Systems, *Yale Univ. Tech. Report* YALEU/DCS/TR-450, December 1985

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

[HM] J. Halpern and Y. Moses, Knowledge and Commmon Knowledge in a Distributed Environment, *Proc. 3rd ACM Symposium on Distributed Computing* 1984 pp. 50-61

[HZ] J. Halpern and L. Zuck, A Little Knowledge goes a Long Way, *Proc. 6th PODC*, 1987, pp. 269-280.

[Lew] D. Lewis, *Convention, a Philosophical Study*, Harvard U. Press, 1969.

[MGM] R. Marvin, M. Greenberg and D. Mossler, "The Early development of conceptual perspective thinking", *Child Development*, **47** (1976) 511-514.

[MT] Y. Moses and M. Tuttle, Programming Simultaneous Actions using Common Knowledge, Research Report MIT/LCS/TR-369 (1987)

[Pa1] R. Parikh, "Knowledge and the Problem of Logical Omniscience", *ISMIS-87*, North Holland, pp. 432-439.

[Pa2] R. Parikh, "Finite and Infinite Dialogues", *Proceedings of a Workshop on Logic and Computer Science*, ed. Moschovakis, Sprinter 1991, 481-98.

[Pa3] R. Parikh, "Social Software", to appear in *Synthese* September 2002.

[PK] R. Parikh and P. Krasucki, "Levels of knowledge in distributed computing", *Sadhana – Proc. Ind. Acad. Sci.* **17** (1992) pp. 167-191.

[PR] R. Parikh and R. Ramanujam, Distributed Computing and the Logic of Knowledge,

Logics of Programs 1985, Springer LNCS 193, 256-268.
[SA] R. Schank and R. Abelson, *Scripts, Plans, Goals, and Understanding*, Erlbaum Hillsdale, NJ (1977)
[SP] C. Steinsvold and R. Parikh, "A Modal analysis of some phenomena in child psychology", *Bulletin of Symbolic Logic*, Mar 2002, Logic Colloquium '01, page 158.
[St] C. Steinsvold, "Trust and other modal phenomena", research report, CUNY Graduate Center, February 2002.
[WP] H. Wimmer and J. Perner, "Beliefs about beliefs: representation and constraining function of wrong beliefs in young children's understanding of deception", *Cognition*, **13** (1983) 103-128.