Local knowledge assertions in a changing world (Extended abstract)

R. Ramanujam

The Institute of Mathematical Sciences
C.I.T. Campus
Madras - 600 113
India
jam@imsc.ernet.in

ABSTRACT

When the state of the world changes due to an action performed by an agent in a multi-agent system, the views of other agents, and hence their knowledge, remain unaffected. We describe such situations using a simple modal logic. Traditionally, modal logics of knowledge are interpreted over *global* states of the multi-agent system. When actions are incorporated into such logics, it leads to high undecidability, whereas, if we see the assertions as made by the agents in the system at their *local* states, we get a decidable logic, for which we also provide a complete axiomatization. Interestingly, when we consider a corresponding local temporal logic of knowledge and (linear) time, the knowledge modality represents a 'necessarily now' modality of present tense.

1 Introduction

In reasoning about knowledge in multi-agent systems, whether it is in the context of artificial intelligence, or distributed computing, or economic theory, it is usual to consider dynamic state spaces alongwith the information partition of agents. Typically, dynamic state spaces contain states indexed by time instants. In the theory of distributed systems, one considers the (infinite) *runs* of a distributed system. Knowledge formulas are interpreted on these runs, or on time-indexed states. Such logics and models have been studied extensively ([HF 89], [HM 90]).

There seems to have been less work on capturing changes in state *due to* actions of agents in the system. While such state changes are usually present in the frames studied, the logical machinery rarely studies how agents' knowledge changes due to actions and conversely how actions cause changes in knowledge states. However, we feel that this is important, because, when the system changes state due to an action by a group of agents in the system, the agents in that group typically know the effect of the action, the knowledge of the rest remains unaffected, and the acting agents know this too.

Two of the TARK '94 papers study such systems : the *environments* of [Me 94] and the *knowledge transition systems* of [KR 94]. However, the logic studied in the former does not include the effect of actions, and the latter concentrates on the temporal structures defined by these systems.

Of particular interest is the case when the set of states is *finite*. Then we can consider the (important) questions of efficiency: how can we compute the knowledge of agents, and how can an agent maintain and update its knowledge ? Once we define an appropriate logic, we can study these questions in terms of the model checking problem for that logic ([HV 91]).

Briefly, knowledge transition systems (KTSs) are state transition systems enriched with n equivalence relations (where n is the number of agents in the system). These equivalence relations, as usual, define which states an agent can distinguish, and interact with transitions in the following manner : when an agent i makes a transition $s \rightarrow s'$, for all other agents j, s and s' are j-equivalent. Further, both the enabling of an action as well as the resulting state are determined only by the views of agents participating in that action. This ensures that when an i-action as well as a j-action are both enabled at a state, they can be "commuted" and the resulting state is the same. (See the next section for more detail.)

[R94] studied a logic of knowledge interpreted over KTSs. Since propositional dynamic logics (PDL) are suited to reason about state changes, $\mathbf{K}_i$ operators were added to a PDL-like logic. Thus the logic had propositional variables, boolean connectives, and modalities for next-state, eventuality and knowledge. Unfortunately, the logic turned out to be highly undecidable, indeed not recursively axiomatizable. This was due to the existence of "commuting" diagrams forced by the frames. (In itself this is no surprise; see [LPRT95] for a variety of negative results of this kind. However, the interesting point is that a seemingly weak next-state modality – unindexed by agents or actions – can force undecidability.)

Can we find a decidable logic of knowledge to reason about KTSs ? It turns out that the solution is to interpret formulas at **local** states of agents rather than on global states of the system, and build up global formulas from local ones. This requires *knowledge assertions at local states.*

This is a departure from the standard fashion, where knowledge is ascribed to agents based on global observations. However, we must hasten to point out that we will continue to use *implicit* knowledge of agents rather than locally computed explicit knowledge, say, as studied by [HMV 94]. Incorporating their notion of algorithmic knowledge into this framework seems to be an interesting question for future study.

Thus, in this paper we will be studying a propositional modal logic of knowledge interpreted on local states of KTSs. However, we will confine our attention to a subclass of KTSs, those in which all communication is only by synchronous handshaking. This is for several reasons : the problem is already quite non-trivial; the model is closely related to concurrent automata ([Zie 87]) studied in the literature; the logic is closely related to the partial order based extensions of temporal logics proposed recently ([T94]). In earlier work [LRT 92], we have studied local reasoning in the context of message passing systems, and related the models to KTSs [KR 94]. Combining those techniques with the approach developed here is left for future work.

In the context of distributed computing, we are more interested in temporal assertions, and hence we study the logic interpreted on runs of the system. This gives us a linear time temporal logic, where assertions are made by agents at their local histories. Since the runs are generated from a finite state system, we are not making assumptions of perfect recall here.

Interestingly, the knowledge modality acts as a present tense modality in this temporal logic. Notice that when temporal assertions are made at global states of a system, the global state corresponds to a time instant (logical rather than physical), and from that instant past and future form an 'event cone' in two directions. The present, referring to a point, is not a modality. Even when time is branching, we can only refer to alternative tense. However, in partial order based models of linear time, each agent sees time as proceeding linearly, but globally time is only partially ordered. Now, a local time instant can correspond to several possible global instants, and an assertion of the form "right now, α necessarily holds for agent j" makes sense when made by an agent $i \neq j$. Thus, we have a local present tense modality 'necessarily now' with its dual 'possibly now', and it is this kind of reasoning embodied by the logic of knowledge presented here.

Thus, we claim that we have an interesting logic of knowledge for the following reasons:

- The frames provide a way of discussing dynamic state spaces, where states change due to actions of agents in the system, which determine and are determined by the agents' knowledge. Such situations often arise in applications of knowledge theory in economics as well as computer science.

- In the logic, we can make assertions like "agent i performs action a at a local state only if i knows that α holds at j". Explicit reasoning about knowledge-based action of this kind seems useful in the context of programs ([HZ 92]).

- The temporal logic suggested has the *trace consistency* property ([Pel 93]): a formula is true along one run of the system if and only if it is true along every interleaving of that run. This means that partial order based techniques ([Val 90]), [GW 94]) leading to fast verification procedures become available for this logic as well.

- The logic is decidable and admits a complete axiomatization. The knowledge modality does not introduce extra complexity in the model checking problem for the temporal logic.

- The modality is philosophically interesting from the viewpoint of tense logics.

We have omitted proofs in this extended abstract to save space, confining ourselves to a brief discussion of techniques used. Our main idea is to introduce the logics here ; the techniques required to prove completeness of the suggested axiomatization as well as decidability are quite involved and will be presented in the full paper. Interestingly, while we can show decidability results for the temporal logic, we have no axiomatization yet, and obtaining one appears to be quite an intriguing problem.

2 The logic and its frames

In this section, we first define knowledge transition systems in the context of agents among whom communication is only by way of synchronization, and then propose the logic for which these are frames.

We assume that the number of agents $n > 0$ is fixed. A *distributed action alphabet* is a pair (Σ, θ), where Σ is a finite nonempty set of actions, and $\theta : \Sigma \rightarrow (2^{\{1,\ldots,n\}} - \emptyset)$. For $a \in \Sigma$, we talk of $\theta(a)$ as the *locations* of a. We can think of agents in $\theta(a)$ jointly performing the action a.

Definition 2.1 *A* **synchronizing knowledge transition system (SKTS)** *over the distributed alphabet* (Σ, θ) *is a tuple* $\mathcal{K} = (S, \rightarrow, \sim_1, ..., \sim_n)$ *, where*

1. S *is a set of* states*,*
2. $\rightarrow \subseteq S \times \Sigma \times S$ *is the* transition relation*,*
3. $\sim_i \subseteq S \times S, i \in \{1, \ldots, n\}$ *are equivalence relations satisfying for all* s, s' *in* S*:*
   (a) *if* $(\forall i, s \sim_i s')$ *then* $s = s'$*.*
   (b) *if* $s \xrightarrow{a} s'$ *and* $i \notin \theta(a)$ *then* $s \sim_i s'$*.*
   (c) *if* $s_1 \xrightarrow{a} s_2$*, and* $\exists s'_1 \in S$ *such that for every* $i \in \theta(a), s_1 \sim_i s'_1$*, then there exists* s'_2 *such that* $s'_1 \xrightarrow{a} s'_2$*, and* $\forall i \in \theta(a), s_2 \sim_i s'_2$*.*

S can be thought of as containing the global states of the system, and $\sim_i$ gives the indistinguishability relation for agent i. Since $[s]_i$, the equivalence class of s under $\sim_i$, specifies all states in which agent i has the same view as in s, we can think of $[s]_i$ as a *local* state of agent i. Note that every transition carries the labels of those agents in the system which participate in that transition: therefore, no external changes are described. In this sense, S represents the states of knowledge of agents rather than the "states of the world".

Condition (a) above is another way of saying that a global state is simply an n-tuple of local states. A local action of agent i cannot alter the knowledge state of another agent j and this is what the second condition specifies. Condition (c) asserts that both enabling of an action and its effect are determined *only* by the knowledge states of agents participating in that action. The latter is a standard implicit assumption in the literature on knowledge in economics, and typical of synchronizing distributed systems.

A word about the communication mechanism here. SKTSs can display a good deal of asynchrony : while agents 1 and 2 are participating in action a, say, agents 3 and 4 could be performing b. Subgroups of agents may be proceeding at different speeds. The reference to synchrony here is only to emphasize that when agents do synchronize, it is by way of handshaking rather than message passing (deals made on telephone rather than via letters). Condition (a) above is reasonable only for such systems. In a message based system, a global state needs to have more information than merely given by a tuple of local states ; for instance, the status of buffers containing undelivered messages needs to be recorded. This can be bypassed by modelling buffers as additional agents, but condition (c) breaks down more crucially. When $s \xrightarrow{a} s'$ and a constitutes the sending of a message by agent 1 to 2, $s \sim_2 s'$ but receipt of the message by 2 is enabled at s' but not in s. Indeed this is why knowledge transition systems are defined in [KR 94] using condition (b) above and the (weaker) first condition below.

Proposition 2.2 *In every SKTS $\mathcal{K} = (S, \rightarrow, \sim_1, ..., \sim_n)$, the following properties hold:*

- *if $s \xrightarrow{a} s_1, s \xrightarrow{b} s_2$ and $\theta(a) \cap \theta(b) = \emptyset$, then there exists s_3 such that $s_1 \xrightarrow{b} s_3$ and $s_2 \xrightarrow{a} s_3$.*
- *if $s \xrightarrow{a} s_1 \xrightarrow{b} s_2$ and $\theta(a) \cap \theta(b) = \emptyset$, then there exists s_3 such that $s \xrightarrow{b} s_3 \xrightarrow{a} s_2$.*

Since dynamic logics use transition systems of the form $(S, \rightarrow, \Sigma)$ as frames, and logics of knowledge are interpreted on systems of the form $(S, \sim_1, \ldots, \sim_n)$, a natural logic to define on SKTSs is a propositional dynamic logic enriched with $\mathbf{K}_i$ modalities. Formally, let $AP = \{p_0, p_1, \ldots\}$ be a countable set of atomic propositions with p ranging over AP. Define the logical language $\mathcal{L}$ as follows:

$\mathcal{L}$ ::= $p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \langle a \rangle \phi \mid \diamond\, \phi \mid \mathbf{K}_i \phi$

(We say "dynamic" in the sense that the eventuality modality steps finitely through an action-indexed next-state modality ; the logic can also be presented using program operators.) Frames for $\mathcal{L}$ are SKTSs and models are pairs $M = (\mathcal{K}, V)$ where $\mathcal{K}$ is a frame and $V : S \rightarrow 2^{AP}$ is a valuation.

The notion 'ϕ holds at state s in model M', denoted $M, s \models \phi$ is defined as usual, by induction on the structure of formulas. Below, we use $\xrightarrow{*}$ to denote the reflexive transitive closure of the $\rightarrow$ relation. The modal cases are defined as follows:

- $M, s \models \langle a \rangle \phi$ iff $\exists s' \in S$ such that $s \xrightarrow{a} s'$ and $M, s' \models \phi$.
- $M, s \models \diamond \phi$ iff $\exists s'$ such that $s \xrightarrow{*} s'$ and $M, s' \models \phi$.
- $M, s \models \mathbf{K}_i \phi$ iff for every $s' \in S$, if $s \sim_i s'$, then $M, s' \models \phi$.

A formula ϕ is said to be *(finitely) satisfiable* if there exists a (finite) model M and a state s in it such that $M, s \models \phi$. A formula is (finitely) valid if and only if $\neg\phi$ is not (finitely) satisfiable.

Unfortunately the logic $\mathcal{L}$ is too expressive : formulas can force models to contain "grids", tilings of the plane. This leads to undecidability. The following theorem [R94] is proved by recursive reductions of instances of colouring problems (variants of tiling problems) to (finite) satisfiability of $\mathcal{L}$ formulas.

Theorem 2.3 *The satisfiability problem for $\mathcal{L}$ is Σ_1^1-complete. The finite satisfiability problem is not recursive either.*

This means that (in both cases) the set of valid formulas is not recursively enumerable, and hence we cannot hope for a finite axiomatization. Thus we have no option but to change the logic.

An analysis of the proof of undecidability shows that the problem is mainly due to the use of *global* propositions and modalities. For a simplified explanation, consider a system with four states s_1, s_2, s_3, s_4 and transitions $s_1 \xrightarrow{a} s_2 \xrightarrow{b} s_4$, $s_1 \xrightarrow{b} s_3 \xrightarrow{a} s_4$, $\Sigma = \{a, b\}$, $\theta(a) = \{1\}, \theta(b) = \{2\}$. The partitions for 1 and 2 are respectively $(\{s_1, s_3\}, \{s_2, s_4\})$, and $(\{s_1, s_2\}, \{s_3, s_4\})$. Consider the valuation V which makes p true at s_2 and s_4 and q true at s_3 and s_4. At s_1, the formula $(\neg p \wedge \neg q) \wedge \langle a \rangle (p \wedge \neg q) \wedge \langle b \rangle (\neg p \wedge q)$ holds, and this is a definite assertion which cannot be made by any agent in the system. Agent 1 can see the change from $\neg p$ to p but would be uncertain about the status of q, and similarly for agent 2. They could pool in their information together, but that would require communication between them which is not possible in this example system. Thus we have a God-like ability to make global assertions, which are not obtained by composing individual knowledge assertions. It is this ability that creates trouble.

We therefore propose a logic where the modalities are *local*, and we compose such assertions globally mainly using boolean connectives. The logical language defined below called $KPDL$ has two levels of syntax, consisting of *local formulas* and *global formulas*. As before, fix AP the set of atomic propositions with p ranging over AP. We use α, β, γ etc. (with or without subscripts) to denote local formulas. The syntax of **i-local formulas** is given by:

$$\Phi_i \ ::= \ p \in P_i \mid \neg\ \alpha \mid \alpha \ \vee \ \beta \mid \langle a \rangle\ \alpha \mid \Diamond\alpha \mid \mathbf{K}\phi$$

where, ϕ is a global formula, defined below.

$$KPDL ::= \alpha @ i,\ \ \alpha \in \Phi_i,\ \ i \in \{1, \ldots, n\} \mid \neg\ \phi \mid \phi_1 \ \vee \ \phi_2 \mid \langle a \rangle \phi,\ Voc(\phi) \subseteq \theta(a)$$

$Voc(\phi)$, the *agent vocabulary* of ϕ is inductively defined in the obvious manner :

$$Voc(\alpha @ i) \stackrel{\text{def}}{=} \{i\};\ Voc(\neg\phi) \stackrel{\text{def}}{=} Voc(\phi);$$

$$Voc(\phi_1 \vee \phi_2) \stackrel{\text{def}}{=} Voc(\phi_1) \cup Voc(\phi_2);$$

$$Voc(\langle a \rangle \phi) \stackrel{\text{def}}{=} \theta(a).$$

At the global level, we have but a restricted modality. This is because, a is a synchronization between agents who participate in a and hence they can exchange their views of the system state. The vocabulary restriction ensures that this is all they get to see. Additional modalities at the global level lead to excessive power, so we stop at this.

A *model* is a tuple $M = (\mathcal{K}, V_1, \ldots, V_n)$, where $\mathcal{K} = (S, \rightarrow, \sim_1, ..., \sim_n)$, and $V_i : S \rightarrow \wp(AP)$ is the i^{th} valuation function such that $s \sim_i s'$ implies $V_i(s) = V_i(s')$. Thus, atomic propositions are evaluated at local states.

The formula ϕ being satisfied in a model M at a state is defined below. We first define the notion for i-local formulas. Let $M_i \stackrel{\text{def}}{=} (L_i, V_i)$, where $L_i \stackrel{\text{def}}{=} \{[s]_i \mid s \in S\}$.

- $M_i, [s]_i \models_i p$ iff $p \in V_i(s)$.
- $M_i, [s]_i \models_i \neg\alpha$ iff $M_i, [s]_i \not\models_i \alpha$.
- $M_i, [s]_i \models_i \alpha \vee \beta$ iff $M_i, [s]_i \models_i \alpha$ or $M_i, [s]_i \models_i \beta$.
- $M_i, [s]_i \models_i \langle a \rangle \alpha$ iff $\exists s_1 \in [s]_i, s_2 \in S$ such that $s_1 \xrightarrow{a} s_2$ and $M_i, [s_2]_i \models_i \alpha$.
- $M_i, [s]_i \models_i \Diamond\alpha$ iff $\exists s_1 \in [s]_i, s_2 \in S$ such that $s_1 \xrightarrow{*} s_2$ and $M_i, [s_2]_i \models_i \alpha$.
- $M_i, [s]_i \models_i \mathbf{K}\phi$ iff for every $s' \in S$, if $s \sim_i s'$, then $M, s' \models \phi$.

Above, $\models$ refers to the global satisfaction relation defined below.

- $M, s \models \alpha@i$ iff $M_i, [s]_i \models_i \alpha$.
- $M, s \models \neg\phi$ iff $M, s \not\models \phi$.
- $M, s \models \phi_1 \vee \phi_2$ iff $M, s \models \phi_1$ or $M, s \models \phi_2$.
- $M, s \models \langle a \rangle \phi$ iff there exists $s' \in S$: such that $s \xrightarrow{a} s'$ and $M, s' \models \phi$.

The derived connectives of propositional calculus such as $\wedge$, $\supset$ and $\equiv$ are defined in terms of $\neg$ and $\vee$ in the usual way. The derived modalities $\Box$, $[a]$ and $\mathbf{L}$ are given by:
$\Box\alpha \stackrel{\text{def}}{=} \neg\Diamond\neg\alpha$; $[a]\phi \stackrel{\text{def}}{=} \neg\langle a \rangle\neg\phi$; $\mathbf{L}\phi \stackrel{\text{def}}{=} \neg\mathbf{K}\neg\phi$.

The notions of satisfiability and validity are as before. Some abbreviations will be convenient: $\langle a \rangle_i \alpha$ will denote $(\langle a \rangle\ \alpha)@i$. Similarly, $\Diamond_i\alpha$ and $\mathbf{K}_i\phi$ will denote the (global) formulas $(\Diamond\alpha)@i$ and $(\mathbf{K}\phi)@i$ respectively.

3 The axiom system

We now present an axiomatization of the valid formulas. We have one axiom system Ax_i for each agent i in the system, and in addition a global axiom system AX to reason about synchronization. In some sense, this helps to isolate how much global reasoning is required.

We use the notation $\vdash_i \alpha$ to mean that the formula $\alpha \in \Phi_i$ is a theorem of system Ax_i. Similarly, $\vdash \phi$ means that ϕ is a theorem of the global system AX.

Before we present the axiom system, a remark about the rules here. Since dynamic logic has an eventuality modality which is obtained by stepping through the next state modality, one needs an induction principle in the logic. The standard form of the rule is like this : (let $\odot$ stand for 'necessarily true in every successor state'.)

$$(Ind)\ \frac{\psi \supset (\alpha \wedge \odot\psi)}{\psi \supset \Box\alpha}$$

Unfortunately, this rule does not work for us, because moves in SKTSs are global and $\Diamond$ refers to reachability at a global level. Hence the premise must be a theorem of the global system, and in the conclusion, we access the global formula via the knowledge modality. But then transporting the premise of the standard form at a global level is not simple, because there is no global next state modality in the logic ! Thus to denote the invariance of ψ, we are forced to examine the decomposition of ψ into local formulas, and ensure that no matter what move b is made, every agent j in $\theta(b)$ locally maintains the invariance of the j^{th} component of ψ. This is what the rule (Ev_i) below says.

Ax_i, The axiom schemes for agent i

$(A0_i)$	All the substitutional instances of the tautologies of PC
$(A1_i)$	$[a](\alpha \supset \beta) \supset ([a]\alpha \supset [a]\beta)$
$(A2_i)$	$\Box(\alpha \supset \beta) \supset (\Box\alpha \supset \Box\beta)$
$(A3_i)$	$\Box\alpha \supset (\alpha \wedge [a]\Box\alpha)$
$(A4_i)$	$\alpha \equiv \mathbf{K}(\alpha@i)$
$(A5_i)$	$\mathbf{K}\phi \supset \mathbf{K}\mathbf{K}_i\phi$
$(A6_i)$	$\neg\mathbf{K}\phi \supset \mathbf{K}\neg\mathbf{K}_i\phi$
$(A7_i)$	$\langle a\rangle True \supset \mathbf{L}\langle a\rangle True$

Inference rules

$$(MP_i)\ \frac{\alpha,\ \ \alpha \supset \beta}{\beta} \qquad (TG_i)\ \frac{\alpha}{[a]\alpha} \qquad (KG_i)\ \frac{\vdash \phi_1 \supset \phi_2}{\mathbf{K}\phi_1 \supset \mathbf{K}\phi_2}$$

Below let ψ be of the form $\bigwedge_k \alpha_k@k$, let ψ_b denote $\bigwedge_{j\in\theta(b)} \alpha_j@j$, and let $next-\psi$ denote $(\bigwedge_{b\in\Sigma} [b]\psi_b))$.

$$(Ev_i)\ \frac{\vdash \psi \supset (\alpha@i \wedge next-\psi)}{\mathbf{K}\psi \supset \Box\alpha}$$

The axioms are quite standard. $(A4_i)$ expresses the fact that the knowledge modality collapses when referring to local formulas. $(A7_i)$ says that a is a joint action performed by all agents in $\theta(a)$ together. (Note that the two $\langle a\rangle True$ formulas in the axiom are quite different ! The former is a local assertion, and the latter is a global one.)

The rule (KG_i) reflects the fact that properties which are invariant in the system (common knowledge properties) are locally known.

We now present the global system. We need to 'tie up' the global next state modality with the local ones : clearly, a conjunction of local $\langle a\rangle$ modalities (over $i \in \theta(a)$) is too weak to force a global a-move in the state where it is asserted. It would merely imply the existence of a-moves in i-indistinguishable states, for each i in $\theta(a)$. Further note that the formula $(\mathbf{K}\alpha@i)@j \wedge \neg\alpha@i$ is not satisfiable, since any system state s satisfying $\neg\alpha@i$ is j-equivalent to itself. This requires a global assertion and suggests axiom $B3$ below.

Global axiom schemes AX

$(B0)$ $(\neg\alpha)@i \equiv \neg\alpha@i$
$(B1)$ $(\alpha \vee \beta)@i \equiv (\alpha@i \vee \beta@i)$
$(B2)$ $[a](\phi_1 \supset \phi_2) \supset ([a]\phi_1 \supset [a]\phi_2)$
$(B3)$ $\mathbf{K}_j\alpha@i \supset \alpha@i$
$(B4)$ $\langle a\rangle(\alpha@i) \supset \langle a\rangle_i\alpha \quad i \in \theta(a)$

Inference rules

$(MP)\ \dfrac{\alpha, \quad \alpha \supset \beta}{\beta} \qquad (GG)\ \dfrac{\vdash_i \alpha}{\alpha@i} \qquad (Gr)\ \dfrac{\vdash \phi, \quad Voc(\phi) \subseteq \theta(a)}{[a]\phi}$

$$(GM) \quad \frac{\bigwedge_{i\in\theta(a)} \alpha_i@i \supset \bigvee_{j\notin\theta(a)} \alpha_j@j}{\langle a\rangle(\bigwedge_{i\in\theta(a)} \alpha_i@i) \supset \bigvee_{j\notin\theta(a)} \alpha_j@j}$$

The rule (GG) allows us to globally infer theorems about agent i from those which have been proved in Ax_i. (For instance, this rule, alongwith (B0) and (B1) allows us to infer "@-versions" of tautologies.) The rule (GM) describes joint moves in the system. In particular, when an a-move is made, this leaves the knowledge of agents outside $\theta(a)$ unchanged, and this is also stated in (GM). This rule typifies the pattern of reasoning in a "true concurrency" based logic. With this rule and (B3), we can infer interesting assertions about the interaction between knowledge and action : for instance, suppose $a \in \Sigma$ such that $i \in \theta(a)$ and $j \notin \theta(a)$; then we can derive the following theorems:

$(C1)$ $\mathbf{K}_j\alpha@i \supset [a]\alpha@i$
$(C2)$ $\langle a\rangle\mathbf{K}_i\alpha@j \supset \alpha@j$

Proposition 3.1 *Every theorem of AX is valid.*

Theorem 3.2 *AX provides a complete axiomatization of the valid formulas of KPDL. Satisfiability of a formula ϕ in KPDL can be decided in nondeterministic exponential time ($2^{O(m)}$, where m is the length of ϕ).*

A standard PDL-style completeness proof does not work here, as the frame conditions, particularly 3(a) and 3(c) in Definition 2.1 can be met only structurally. Conditions 3(a) and 3(b) are met by carrying global states as n-tuples of local states and using the rule (GM) for making moves. However, condition 3(c) requires that the effect of an action a is determined only by the states of agents in $\theta(a)$, and this is far from easy: a $\mathbf{K}_j$ formula, where $j \notin \theta(a)$ can constrain the resulting state, and this is where axiom (B2) proves to be crucial. On the other hand, an $\mathbf{L}_i$ requirement cannot be simply met by adding a new i-equivalent state; the new one may violate condition 3(c). Ensuring this while working with subformulas of the given formula makes the exercise quite interesting. Further, if we work with n-tuples of sets of formulas, the complexity becomes $2^{O(mn)}$ where m is the length of the formula being decided, and n is the number of agents. We can restrict ourselves to agents mentioned in the given formula but this would still mean a $2^{O(m^2)}$ complexity. Therefore, given an AX-consistent formula, we first construct a pseudomodel of size $2^{O(m)}$ which violates conditions 3(a) and 3(c) and then expand it out to a model. However, a formula is satisfiable if and only if it is pseudo-satisfiable, and that gives the result.

4 Temporal logic

In this section, we study a linear time temporal logic defined on the runs of an SKTS. For this, we confine our attention to rooted systems. A *rooted* SKTS is a pair $(SKTS, s_0)$, such that $s_0 \in S$ and every state in S can be reached from s_0 by a path in $\rightarrow$.

The frames for the logic will be non-sequential runs of SKTSs. A run of a system is normally defined as a sequence of transitions, but this not only resolves choices in the system, but also loses any information about concurrency. If disjoint groups of agents perform actions a and b at a state, then these actions are concurrent. The idea of a nonsequential run is precisely to capture this while resolving choices. For this, we need to look at the *unfoldings* of SKTSs, which contain all the system runs. There is a systematic way of doing this (in a manner similar to that done in [KR 94] for message systems), yielding what we call *synchronization structures*, introduced in [R 95]. However, we will need only a subclass of these structures here. (These are simply called *traces* in [T94].)

Definition 4.1 *A* **Linear Synchronization Structure** *with n agents (an n-SS) over a distributed alphabet (Σ, θ) is a triple $\mathcal{E} = (E, \leq, \lambda)$, where*

(i) *E is a set of event occurrences,*

(ii) *$\leq \subseteq E \times E$ is a partial order called the causality relation such that $\forall e\{e' | e' \leq e\}$ is finite,*

(iii) *$\lambda : E \rightarrow \Sigma$ is a labelling function such that $\forall i \in \{1, ..., n\}$, E_i is totally ordered by $\leq$, where $E_i \stackrel{\text{def}}{=} \{e' \in E \mid i \in \theta((\lambda(e'))\}$, and*

(iv) *$\leq \; = \; (\bigcup_i \leq_i)^*$, where $\leq_i \stackrel{\text{def}}{=} \; \leq \bigcap (E_i \times E_i)$.*

We will use the notation $\downarrow e$ for the set $\{e' | e' \leq e\}$; similarly for $X \subseteq E$, $\downarrow X = \{e' | \exists e \in X, e' \leq e\}$. We will also speak of SSs, leaving n implicit.

We say that e_1 and e_2 are *concurrent* if $\theta(\lambda(e_1)) \cap \theta(\lambda((e_2)) = \emptyset$, and e_1 and e_2 are incomparable under the causal ordering. Note that for any $e_1, e_2 \in E$, we have $e_1 \leq e_2$ or $e_2 \leq e_1$ or e_1 and e_2 are concurrent.

We can now define a notion of a *configuration* in an SS: $c \subseteq E$ is a configuration iff $\downarrow c \subseteq c$. Note that the empty set is always a configuration. More importantly, for any $e \in E$, the set $\downarrow e$ is a configuration, a fact which we will use crucially later on. For an SS, let C_{SS} denote the set of its *finite* configurations. Further, let $L_i^{SS} \stackrel{\rm def}{=} \{\emptyset\} \cup \{\downarrow e |\ e \in E_i\}$ denote the set of *i-local configurations.*

We can define a transition relation on configurations as follows: $c \stackrel{e}{\rightarrow} c'$ iff $e \notin c$ and $c' = c \cup \{e\}$. Similarly, we can define n equivalence relations by: $c \sim_i c'$ iff $c \cap E_i = c' \cap E_i$.

Definition 4.2 *A* **frame** *is a tuple $F = (\mathcal{K}, \mathcal{E}, \mu)$, where $\mathcal{K} = (S, \rightarrow, \sim_1, \ldots, \sim_n)$ is a* finite *SKTS over (Σ, θ) rooted at $s_0 \in S$, $\mathcal{E} = (E, \leq, \lambda)$ is an SS over the same alphabet, and $\mu : C_{SS} \rightarrow S$ is a map such that $\mu(\emptyset) = s_0$, whenever $c \stackrel{e}{\rightarrow} c'$ and $\lambda(e) = a$, we also have $\mu(c) \stackrel{a}{\rightarrow} \mu(c')$ in the SKTS, and whenever $c \sim_i c'$, we also have $\mu(c) \sim_i \mu(c')$ in the SKTS.*

It is easy to see that a frame represents a non-sequential run of the SKTS : that is, a run and all other interleavings of the same run (when disjoint groups of agents perform actions a and b at a state, a run arriving at that state and proceeding via the sequence ab is different from another which is the same till that state and proceeds via ba; however these two runs are but different interleavings of the same run).

We now present the logical language called $KPTL$, which has almost the same syntax as $KPDL$, except that we no longer need a global next-state modality. The reason for this should be clear : since the frames are runs, at any state there can at most be one a-successor, for any a, and hence the conjunction of local next-state modalities is equivalent to a global next-state modality, unlike in the dynamic logic. It is also more appropriate to have locally an *until* modality rather than $\Diamond$ as before. Formally, the syntax of $KPTL$ is given as follows:

$$\Psi_i \ ::= \ p \ \in \ P_i \mid \neg\ \alpha \mid \alpha \ \vee \ \beta \mid \langle a \rangle\ \phi, \ a \in \Sigma_i, \ Voc(\phi) \subseteq loc(a) \mid \alpha \mathbf{U} \beta \mid \mathbf{K}\phi$$

$$KPTL ::= \alpha@i, \ \alpha \in \Psi_i, \ i \in Loc \mid \neg\ \phi \mid \phi_1 \ \vee \ \phi_2$$

$$Voc(\alpha@i) \stackrel{\rm def}{=} \{i\}; \ Voc(\neg\phi) \stackrel{\rm def}{=} Voc(\phi); \ Voc(\phi_1 \vee \phi_2) \stackrel{\rm def}{=} Voc(\phi_1) \cup Voc(\phi_2)$$

A *model* is a tuple $M = (F, V_1, \ldots, V_n)$, where F is a frame, and $V_i : S \rightarrow \wp(AP)$ is the i^{th} valuation function such that $s \sim_i s'$ implies $V_i(s) = V_i(s')$.

The formula ϕ being satisfied in a model M at a configuration is defined below. We first define the notion for i-local formulas. These are to be interpreted in the local configurations. Let $M_i \stackrel{\rm def}{=} (L_i^{SS}, V_i)$. Let ρ, ρ' range over i-local configurations.

- $M_i, \rho \models_i p$ iff $p \in V_i(\mu(\rho))$.
- $M_i, \rho \models_i \neg\alpha$ iff $M_i, \rho \not\models_i \alpha$.
- $M_i, \rho \models_i \alpha \vee \beta$ iff $M_i, \rho \models_i \alpha$ or $M_i, \rho \models_i \beta$.
- $M_i, \rho \models_i \langle a \rangle \phi$ iff $\exists e \in E_i - \rho$, $\exists c, c' \in C_{SS}$ such that $\lambda(e) = a$, $\rho \sim_i c$, $c \stackrel{e}{\rightarrow} c'$, and $M, c' \models \phi$.
- $M_i, \rho \models_i \alpha \mathbf{U} \beta$ iff $\exists \rho'$ such that $\rho \subseteq \rho'$, $M_i, \rho' \models_i \beta$ and $\forall \rho'' : \rho \subseteq \rho'' \subset \rho'$, $M_i, \rho'' \models_i \alpha$.

- $M_i, \rho \models_i \mathbf{K}\phi$ iff for every configuration $c \in C_{SS}$, if $c \sim_i \rho$, then $M, c \models \phi$.

It can be easily seen that this is a linear time temporal logic, augmented with a knowledge modality which refers to global properties invariant with respect to the view of agent i at its local configuration. The $\langle a \rangle$ modality is similar to the one in $KPDL$ but the difference is that now it refers to the next local state within the run. The derived connectives and modalities are as before. We now define the semantics of global formulas. For a configuration c, let the *i-view* at c be given by: $\downarrow_i(c) \stackrel{\text{def}}{=} \downarrow(c \cap E_i)$. Note that this is in L_i^{SS}.

- $M, c \models \alpha@i$ iff $M_i, \downarrow_i(c) \models_i \alpha$.
- $M, c \models \neg\phi$ iff $M, c \not\models \phi$.
- $M, c \models \phi_1 \vee \phi_2$ iff $M, c \models \phi_1$ or $M, c \models \phi_2$.

We say that a formula ϕ is satisfiable if there exists a model M such that $M, \emptyset \models \phi$. Note that this logic is trivially seen to have the trace consistency property : a global formula is satisfied in one run of the SKTS if and only if it is satisfied along every interleaving of the same run. It can be easily checked that every such set of trace equivalent runs generates a linear synchronization structure and a map μ from its configurations to the states of the SKTS, and vice versa.

Theorem 4.3 *The satisfiability problem for $KPTL$ is decidable in nondeterministic exponential time ($2^{O(m)}$, where m is the length of the formula to be decided).*

The main difficulty of the proof lies in isolating a subset of global states as those of the form $\downarrow e$. Such a state is important, because when $e \in E_i$, this is the smallest configuration in the i-equivalence class, and hence an $\mathbf{L}_i$ requirement can be satisfied in its 'future'. Unlike in the earlier logic, an $\mathbf{L}_i$ formula here refers to configurations *within the same run* (and hence acts as "possibly now"), and this considerably complicates the proof.

The proof we have is automata theoretic : for every formula we construct a nondeterministic Büchi automaton which accepts a nonempty language if and only if that formula is satisfiable. The main idea is to carry along $\mathbf{L}$ requirements in the automaton and switch whenever we reach states corresponding to local configurations.

For this logic we can define the *model checking problem* as follows: given a finite $SKTS$, a valuation and a formula ϕ_0, is there a non-sequential run given by $(\mathcal{E}, \mu)$ and a model based on it using the given valuation such that ϕ_0 is satisfied at the initial state ?

Theorem 4.4 *The model checking problem for $KPTL$ is decidable in time $k \times 2^{O(m)}$, where k is the number of states in the given $SKTS$ and m is the length of the formula to be checked.*

(The proof of this is an easy extension of the earlier automata theoretic one. The technique ensures that we also solve the problem posed differently : given a finite $SKTS$, a valuation and a formula, check whether the formula is satisfied in *every* nonsequential run of the system.)

Unfortunately, these techniques do not guide attempts at axiomatization. It is clear that we need to add some rules. For instance, if agents 1 and 2 can synchronize on a, the formula $\langle a \rangle_1 True \wedge \Box_2 [a] False$ is no longer satisfiable. This can be handled by a new rule similar to (Ev) : it says that when one party to a synchronization commits to synchronizing, all parties eventually "get to the table".

Below let $\psi \stackrel{\mathrm{def}}{=} \bigwedge_k \alpha_k @k$, and let ψ_b denote $\bigwedge_{j \in \theta(b)} \alpha_j @j$, as before. Further let $\widehat{i}-next-\psi$ denote $\bigwedge_{b \in \Sigma, i \notin \theta(b)} \bigwedge_{j \in \theta(b)} [b]_j \psi_b$.

$$(Sy_i) \quad \frac{\vdash \psi \supset ((\bigvee_{j \in \theta(a)} [a]_j False) \wedge \widehat{i} - next - \psi)}{\mathbf{K}\psi \supset [a] False}$$

In fact, with this rule, and additional global propositions (to denote i-local configurations) we can obtain a completeness theorem, but that is a messy solution. What we would like is to axiomatically isolate the i-local configurations, and that seems difficult.

Acknowledgement: I thank Kamal Lodaya for stimulating discussions on various notions of knowledge and tense.

References

[GW94] Godefroid, P. and Wolper, P., "A partial approach to model checking", *Information and Computation*, vol 110, 1994, 305-326.

[HF89] Halpern, J., and Fagin, R., "Modelling knowledge and action in distributed systems", *Distributed Computing*, vol 3, #4, 1989, 159-177.

[HM90] Halpern, J., and Moses, Y., "Knowledge and Common Knowledge in a Distributed Environment", *JACM*, vol. 37, #3, 1990, 549-587.

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

[HV91] Halpern, J., and Vardi, M., "Model checking vs theorem proving: a manifesto", in *Artificial Intelligence and Mathematical Theory of Computation*, (ed) V. Lifschitz, Acad Press, 1991, 151-176.

[HZ92] Halpern, J., and Zuck, L., "A little knowledge goes a long way: simple knowledge-based derivations and correctness proofs for a family of protocols", *JACM*, vol. 39, #3, 1992, 449-478.

[KR94] Krasucki, P., and Ramanujam, R., "Knowledge and the ordering of events in distributed systems", *TARK V*, Theoretical Aspects of Reasoning about Knowledge, 1994, 267-283.

[LPRT95] Lodaya, K., Parikh, R., Ramanujam, R., and Thiagarajan, P.S., "A logical study of distributed transition systems", *Inf. and Comp.*, vol. 19, #1, 1995, 91-118.

[LRT92] Lodaya, K., and Ramanujam, R., and Thiagarajan, P.S., "Temporal Logics for Communicating Sequential Agents: I", *Int. J. Found. Comp. Sci.*, vol. 3, #2, 1992, 117-159.

[Mey94] van der Meyden, R., "Common knowledge and update in finite environments I", *TARK V*, Theoretical Aspects of Reasoning about Knowledge, 1994, 225-242.

[Pel93] Peled, D., "All from one and one from all: on model checking using representatives", *LNCS 697*, 1993.

[R94] Ramanujam, R., " Knowledge and the next state modality", *Indian National Seminar in TCS*, 1994, 62-80.

[R95] Ramanujam, R., "A local presentation of synchronizing systems", *Proc. STRICT, Berlin*, Springer-Verlag, 1995.

[T94] Thiagarajan, P.S., "A trace based extension of propositional linear time temporal logic", *Proc IEEE LICS*, 1994, 438-447.

[Val90] Valmari, A., "A stubborn attack on state explosion", *LNCS 531*, 1990, 156-165.

[Zie87] Zielonka, W., "Notes on finite asynchronous automata", RAIRO-Inf. Theor. et Appli., vol 21, 1987, 99-135.