Reasoning about Events and Knowledge in Distributed Systems

Carla A. D. M. Delgado*

Mario R. F. Benevides*†
COPPE-Sistemas*
Instituto de Matemática†
Universidade Federal do Rio de Janeiro (UFRJ)
PoBox 68.511, ZIP Code 21.945-970, Rio de Janeiro, RJ, Brazil
{delgado, mario}@cos.ufrj.br

Abstract *This work focuses on the handling of knowledge and time in distributed asynchronous systems using logics, specifically on fully asynchronous distributed memory systems. An event-based logic for no-fault asynchronous systems is presented, and a set of event-based temporal operators are defined providing a greater temporal expressivity to the logic, even considering time under the few restrictions of asynchronous model. An application involving knowledge in distributed asynchronous systems was modelled using this logic, illustrating its expressivity and applicability.*

Keywords: distributed systems, modal logic, knowledge representation

1 Introduction

The research in modal logic for knowledge representation has grown in the last two decades mainly due to the work of Halpern, Fagin, Moses and Vardi [1]. However, the application of their logic takes place in the context of the synchronous systems, having simultaneity as the basis for concepts like common knowledge.

In [3], an extension of [1] with temporal operators is presented. This new logic allows for expressing and reasoning about the dynamic of knowledge evolution in time. Unfortunately, this approach is more suitable for reasoning about knowledge in time in distributed synchronous systems.

Reasoning about the tasks in a distributed system at knowledge level offers some advantages like abstracting from system's implementation details or even from agent's nature. A formal approach at knowledge level allows for analyzing the problem's properties before implementing it. On this sense, temporal logics have proved to be useful for specifying concurrent systems, because they can describe ordering of events in time without introducing time explicitly.

The main contribution of this paper is the presentation of a logical model to represent knowledge and time in an asynchronous distributed system where agents communicate by message-passing. The model is capable of representing relations between knowledge in a group of agents and relations between knowledge and time in the event-based model for asynchronous systems of [2].

The paper is organized as follows. Section 2 presents a model for asynchronous distributed systems. In section 3, we define our language and in section 4 its formal semantics. Section 5 presents an application and section 6 contains concluding remarks and future work.

2 System Model

This section introduces all the concepts needed about Asynchronous Distributed Systems (ADS). A Distributed System is a system composed of a set of agents that do not share any memory and can communicate only by sending and receiving messages along a previously defined network. For an asynchronous distributed system, there is no global clock and the delivery time of messages is finite but unbounded. We will introduce an event-based formalism to describe the distributed computations that take place in ADS, according to the model described in [2].

Consider a simple model for an asynchronous distributed system: a network with m agents, connected by FIFO channels; a set R of asynchronous runs (distributed computations or parallel run of all agents involved); a set E of events (agent i sends/receives a message); a set C of cuts (or global states) of the system and a protocol P (or distributed algorithm) that specifies the actions each agent takes in response to receiving a message. A "send message" event generated by an agent implies a "receive message" event at the target agent. A run can be thought as a set of events Ξ. Σ_i is the state sequence that an agent n_i goes through as Ξ evolves. We will define temporal relations on the set of events as follows:

Temporal relation $\prec$: consider two events v_1 and v_2, $v_1 \prec v_2$ if and only if:

- Both v_1 and v_2 occurs at the same node, respectively at instants t_1 and t_2 so that $t_1 < t_2$. No event v' occurs at the same node at an instant t so that $t_1 < t < t_2$.
- The events v_1 and v_2 occurs respectively in processes n_i, n_j so that a message j is sent from n_i to n_j in v_1 and received by n_j in v_2.

The meaning of $\prec$ is "v_1 happened immediately before v_2", and only makes sense if v_1 and v_2 are events on the same run.

Temporal relation $\prec^+$: is the transitive and irreflexive closure of $\prec$. $\prec^+$ establishes a partial ordering over the set of events Ξ. Two events v_1 and v_2 which are not ordered by $\prec^+$ ($(v_1, v_2) \in \Xi\times\Xi - \prec^+$ and $(v_2, v_1) \in \Xi\times\Xi - \prec^+$) are called concurrents.

We can use the relation $\prec^+$ to define the future and past of an event ξ in relation to run Ξ: $Past(\xi) = \{\xi' \in \Xi \,|\, \xi' \prec^+ \xi\}$ and $Future(\xi) = \{\xi' \in \Xi \,|\, \xi \prec^+ \xi'\}$.

System State: is a collection of local states, one for each node, and an "edge state" for each network channel.

Consistent Global State or Global State: in order to define a consistent global state, it's necessary to establish a total order $<$ over Ξ consistent with $\prec^+$. Pairs of consecutive events $(\xi_1, \xi_2) \in <$ if for all event $\xi \neq \xi_1, \xi_2$ either $\xi < \xi_1$ or $\xi_2 < \xi$. There is a system state associated to each pair (ξ_1, ξ_2) of consecutive events in $<$ denoted by $system\_state(\xi_1, \xi_2)$ with the following properties:

- for a node n_i, σ_i is the resulting state from the occurrence of the most recent event in n_i, for example ξ, which is not the case that $(\xi_1 > \xi)$.
- for each channel $(n_i, n_j)\Phi_{ij}$ is the set of messages sent in connection with an event ξ so that it is not the case that $(\xi_1 > \xi)$ and received in connection with an event ξ' so that it is not the case $(\xi' > \xi_2)$.

A system state Ψ is global if and only if either all agents are in their initial states and all channels are empty, or all agents are in their final states and all channels are empty, or there is a total order $<$ consistent with $\prec^+$ for which there is a pair (ξ_1, ξ_2) of events s that $\Psi = system\_state(\xi_1, \xi_2)$.

Another definition of global state, using partitions on the set of events can be seen in [2]. According to it, a system state Ψ is global if and only if it is represented by a partition (Ξ_1, Ξ_2) of Ξ so that: $Past(\xi) \subseteq \Xi_1$ every time that $\xi \subseteq \Xi_1$ or $Future(\xi) \in \Xi_2$ every time that $\xi \in \Xi_2$. The partitions that follow this restriction are called consistent cuts (figure 1).

Future and past of a global state Ψ are defined as: $Past(\Psi) = \bigcup_{\xi\in\Xi_1}[\{\xi\} \cup Past(\xi)]$ and

$Future(\Psi) = \bigcup_{\xi \in \Xi_2}[\{\xi\} \cup Future(\xi)]$ ($\Psi = system\_state(\Xi_1, \Xi_2)$).

We say that a global estate Ψ_1 comes before another global state Ψ_2 in a computation Ξ if and only if: $Past(\Psi_1) \subset Past(\Psi_2)$ or $Future(\Psi_2) \subset Future(\Psi_1)$.

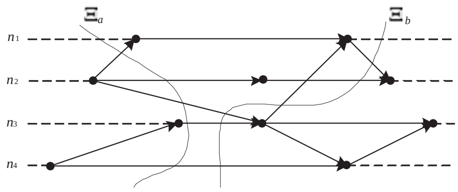

Figure 1: A precedence graph with two partitions: the first is a consistent cut, whereas the second is not

The past view of an agent n_j considering the global state s given by the partition (Ξ_1, Ξ_2) is the set if events that occurs to process n_j in the past of ξ_j, where ξ_j is the event associated with n_j's local state: $Vp(n_j, s) = \{\xi'_j \in \Xi_1 \,|\, \xi'_j \prec^+ \xi_j\}$. Future view can be defined following the same reasoning.

3 The language

We now present a modal language for multiple agents that propose a formal approach to the representation of knowledge and time in ADS, the Event-based Knowledge Language (EKL). As events are the base unit of time for ADS, our language must be capable of dealing with such concept. We are also interested on expressing properties about knowledge during an ADS run. To achieve that, we use epystemic modal operators like the ones in [1].

Operators: the operators are as following:

- The booleans connectives as the ones in propositional classical logic;
- The modals "K_i" and "B_i" follows the definitions for the knowledge and belief logic ([1]);
- The modal "$[v_i]$" represents the validity after the occurrence of event v_i, for $i = 1, ..., m$;
- The modal "$< v_i >$" represents the eventual validity after the occurrence of event v_i, for $i = 1, ..., m$; i. e.: in a moment after the occurrence of v_i, φ will always be valid;
- The modal "(v_i)" represents the validity immediately after the occurrence of event v_i, for $i = 1, ..., m$ (what does not mean that φ cannot became false later);
- The modal "$\Box$" represents the validity after the current estate;
- The modal "$\diamond$" represents possibility after the global estate.
- The modal "$UNTIL$" represents the conditional validity relation between a formula and an event, this way: $\varphi \, UNTIL \, v_i$ indicates that φ holds until the event v_i happens.

4 Semantics

In order to give a Kripke interpretation to knowledge and events modalities it is necessary to establish an appropriate set of possible worlds and relations among them. To express the knowledge notion we use the "possibility relation" over global states as defined in [1]. To express knowledge over time we establish partial temporal relations over global states. An ADS can be considered as a Frame for a Kripke structure, where the possible words are the consistent cuts or global states, the basic facts are the primitives and the relations among global states are described as follows:

Possibility relation based on the past view ($\sim_i$): Defined over the consistent cuts, using the concept of "indistinguishable states". Two global estates or consistent cuts s and s' are indistinguishable in relation to agent $n_i (s \sim_i s')$ if n_i has the same past view in

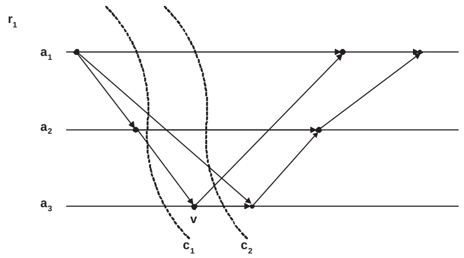

Figure 2: The global states E_1 and E_2 determined by the cuts c_1 and c_2 differs by the occurrence of event v to agent a_3. As a_1 and a_2 can't note that event: $E_1 \sim_1 E_2$ and $E_1 \sim_2 E_2$

s and s'. The possibility relation is reflexive, transitive and symmetric.

Event accessibility relation (R_v^i): We say that a global state s is accessible from other global state t by event v if the occurrence of v on state s transforms the global state from s to t. The relation R_v^i will be defined as follows: "two global states are related by the event accessibility relation R_v^i) if and only if they differ only by the occurrence of one event to agent i". R_v^i is a temporal relation, because ordering global states according to the occurrence of an event corresponds to establish which state precedes other in a computation. This ordering will always be partial, given the properties of the asynchronous model. To formalize R_v^i definition the first definition of global state will be used:

Two global states s e s' relates by R_v^i in relation to an event v_i (noticed by agent n_i) if and only if there is a total order $<$ consistent with $\prec^+$ so that:

- s is a initial state and $Past(v_i)$ is empty (v_i is the first event that occurred to n_i) and $s' = system\_state(v_i, v_2)$ (to some event v_2 according to $<$);

- s' is a final state and $Future(v_i)$ is empty (v_i is the last event that occurred to n_i) and $s = system\_state(v_1, v_i)$ (to some event v_1 according to $<$);

- $s = system\_state(v_1, v_i)$ and $s' = system\_state(v_i, v_2)$ according to the same total ordering. v_1, v', v_2 are consecutive events in such order $<$.

According to this definition, all agents $n_k \neq n_i$ stay at the same local state in s' that they were in s. The state of the edges that do not come from $n_i(\Phi_{kj}$ where $k \neq i)$ also stay the same. The state of the edges Φ_{ij} that can be modified in s' as a consequence of the message occasioned by event v_i. In figure 2 agent a_3 notes the occurrence of v. The global states E_1 and E_2 determined by cuts c_1 and c_2 differs by the occurrence of v to a_3, so: $E_1 R_v^3 E_2$. Two states s and t are related by relation R_v^i, $sR_v^i t$, if t succeeds s at the run. Relation R_v^i is irreflexive and anti-symmetric.

It is interesting to note the "complementary" character of the relations $\sim_i$ and R_v^i. While $\sim_i$ relates global states where the local state of agent n_i is the same, R_v^i relates states where the local states of agent i are different. This implies that states related by $\sim_i$ do not relate by R_v^i, what can be noticed in figure 2 . Let s and s' be two consistent global states. The following formulas holds for every $i, j \mid j \neq i$: $s \sim_i s' \to \neg(sR_v^i s')$, $sR_v^i s' \to \neg(sR_v^j s')$, $sR_v^i s' \to s \sim_j s'$.

Local temporal event based relation R_+^i: For each agent i, we can define a relation R_+^i based on relation R_v^i and $\sim_i$ as: $R_+^i = [\sim_i \circ \cup (R_v^i) \circ \sim_i]^+$. Intuitively, R_+^i represents the transitive closure of the composition of R_v^i and $\sim_i$ for every event. The relation R_+^i provides a temporal order over global states from an agent's point of view.

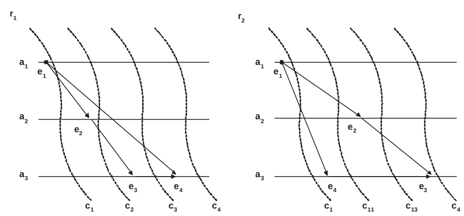

Figure 3: Precedence Graph to initial events of two distinct runs of the Propagation Information with Feedback algorithm (PIF). The initial global state c_0 is not represented

Bellow, we show relations $\sim_i$, R_v^i and R_+^i for two partial runs shown in figure 3:

Run r_1, Agent a_1:

$\sim_1=$	$\{(c_0,c_0),(c_1,c_1),(c_2,c_2),(c_3,c_3),(c_4,c_4),(c_1,c_2),(c_1,c_3),(c_1,c_4),(c_2,c_1),(c_2,c_3),(c_2,c_4),(c_3,c_1),(c_3,c_2),(c_3,c_4),(c_4,c_1),(c_4,c_2),(c_4,c_3)\}$
$R_{e1}^1 =$	$\{(c_0,c_1)\}$
$R_+^1 =$	$\{(c_0,c_1),(c_0,c_2),(c_0,c_3),(c_0,c_4)\}$

Run r_1, Agent a_2:

$\sim_2=$	$\{(c_0,c_0),(c_1,c_1),(c_2,c_2),(c_3,c_3),(c_4,c_4),(c_0,c_1),(c_1,c_0),(c_2,c_3),(c_2,c_4),(c_3,c_2),(c_3,c_4),(c_4,c_2),(c_4,c_3)\}$
$R_{e2}^2 =$	$\{(c_1,c_2)\}$
$R_+^2 =$	$\{(c_1,c_2),(c_0,c_2),(c_1,c_3),(c_1,c_4),(c_0,c_3),(c_0,c_4)\}$

Run r_1, Agent a_3:

$\sim_3=$	$\{(c_0,c_0),(c_1,c_1),(c_2,c_2),(c_3,c_3),(c_4,c_4),(c_0,c_1),(c_0,c_2),(c_1,c_0),(c_1,c_2),(c_2,c_0),(c_2,c_1)\}$
$R_{e3}^3 =$	$\{(c_2,c_3)\}$
$R_{e4}^3 =$	$\{(c_3,c_4)\}$
$R_+^3 =$	$\{(c_2,c_3),(c_0,c_3),(c_1,c_3),(c_3,c_4),(c_2,c_4),(c_0,c_4),(c_1,c_4)\}$

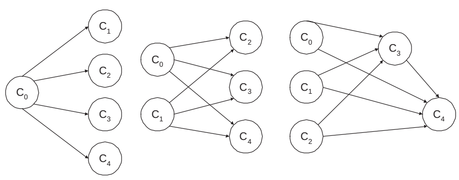

Figure 4: Graphs representing relations R_+^1, R_+^2 and R_+^3 over the global states of run r_1, respectively

For each agent i R_+^i describes how its local state evolves as events occur on time. As we can have many possibilities to consistent cuts, the relation reflects the various possible successors of a state. R_+^i is irreflexive, transitive and anti-symmetric.

Global temporal event based relation $R_\cup$: To represent the notion of global time, we'll define a global temporal event based relation as the union of R_v^i for every event v_i, for every agent i of the group. We'll call it $R_\cup$, $R_\cup = R_v^1 \cup R_v^2 \cup ... \cup R_v^n$. (See figures 6 and 7)

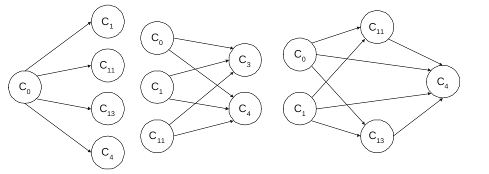

Figure 5: Graphs representing relations R_+^1, R_+^2 and R_+^3 over the global states of run r_2, respectively

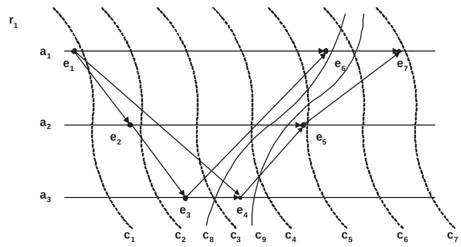

Figure 6: Precedence graph for one run of PIF in an ADS with 3 agents

$R_\cup$ for PIF run in figure 6:

$R_{e1}^1 =$	$\{(c_0,c_1)\}$	$R_{e7}^1 =$	$\{(c_6,c_7)\}$
$R_{e6}^1 =$	$\{(c_3,c_8),(c_4,c_9),(c_5,c_6)\}$		
$R_{e2}^2 =$	$\{(c_1,c_2)\}$	$R_{e5}^2 =$	$\{(c_4,c_5),(c_9,c_6)\}$
$R_{e3}^3 =$	$\{(c_2,c_3)\}$	$R_{e4}^3 =$	$\{(c_3,c_4),(c_8,c_9)\}$
$R_\cup =$	$\{(c_0,c_1),(c_1,c_2),(c_2,c_3),(c_3,c_4),(c_3,c_8),(c_4,c_5),(c_4,c_9),(c_8,c_9),(c_5,c_6),(c_9,c_6),(c_6,c_7)\}$		

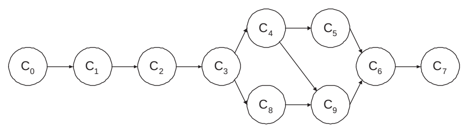

Figure 7: Graph of relation $R_\cup$ over the global states of run of PIF shown in figure 6

For each set of consistent cuts of a run, $R_\cup$ gives a temporal ordering of the states (and consequently, of the events). As it's not possible to define the exact ordering, $R_\cup$ is not a linear order. In figures 6 and 7, state c_3 can be succeeded by c_4 or c_8, what can be seen by the presence of pairs (c_3,c_4) and (c_3,c_8) in $R_\cup$. $R_\cup$ is irreflexive and anti-symmetric.

We are now ready to define Frame and Model for EKL:

Frame:

A frame $F = (S, \sim_i, R_v^i, R_+^i, R_\cup), i = 1, ..., m$

is a structure where:

- S is the set of states or possible worlds;
- $\sim_i, R_v^i, R_+^i, R_\cup$ are binary relations in $S, i = 1, ..., m (\sim_i \subseteq S \times S, R_v^i \subseteq S \times S, R_+^i \subseteq S \times S, R_\cup \subseteq S \times S)$;

Model: A model M over $F = (S, \sim_i, R_v^i, R_+^i, R_\cup)$ is a pair $M = (F, \pi)$, where π is an interpretation that associates truth values to the primitives on Φ on each state from $S, \pi : \Phi \times S \to \{true, false\}$.

Satisfiability: A formula φ is true in (M, s), (in a state $s \in S$) for a model M when:

- $M, s \models p$ if and only if $\pi(s, p) = true$, where $p \in \Phi$;
- $M, s \models \neg\varphi$ if and only if it's not the case that $M, s \models \varphi$;
- $M, s \models \varphi \wedge \psi$ if and only if $M, s \models \varphi$ and $M, s \models \psi$;
- $M, s \models K_i\varphi$ if and only if for all $t \in S$ so that $(s, t) \in \sim_i, M, t \models \varphi$;
- $M, s \models B_i\varphi$ if and only if there exists t so that $t \in S$ e $(s, t) \in \sim_i$, and $M, t \models \varphi$;
- $M, s \models [v_i]\varphi$ if and only if for all states that succeed the happening of v_i holds φ, or: for all pair of states s', s'' so that $s' R_v^i s''$ where $s' \in Future(s), M, s'' \models \varphi$, and for all state s''' so that $s'' R_+^i s''', M, s''' \models \varphi$.
- $M, s \models < v_i > \varphi$ if and only if there exists some state that succeeds the happening of v_i from which φ always hold, that is, exists a pair of states s', s'' so that $s' R_v^i s''$ where $s' \in Future(s)$ and: $M, s'' \models \varphi$ and for all s''' where $s'' R_+^i s'''$ holds $M, s''' \models \varphi$, or exists a state s''' so that $s'' R_+^i s'''$ and $M, s''' \models \varphi$, and for all s'''' so that $s''' R_+^i s'''', M, s'''' \models \varphi$.
- $M, s \models (v_i)\varphi$ if and only if there exists s' so that $s R_v^i s'$ and $M, s' \models \varphi$.
- $M, s \models \Box\varphi$ if and only if for all s' so that $s R_\cup s'$ and $M, s' \models \varphi$.
- $M, s \models \Diamond\varphi$ if and only if there exists s' so that $s R_\cup s'$ and $M, s' \models \varphi$.
- $M, s \models \varphi UNTIL v_i$ if and only if for all states that succeed s and precedes the happening of v_i, φ holds: for all pair of states s', s'' so that $s' R_v^i s''$ where $s' \in Future(s)$, for all state s''' so that $s''' \in Future(s)$ and $s''' \in Past(s'), M, s''' \models \varphi$.

Valid formulas

We now present some valid formulas and inference rules on the logic defined.

Formulas 1 to 3 says that any of the necessity modal operators of the logic can be distributed over implication, formulas 4 to 6 presents the dual modalities, formulas 7 to 9 represents the hierarchy over the modal operators that deal with time, formulas 10 and 11 shows how knowledge can be spread over time for each one of the temporal operators and 12 to 14 shows inference rules.

1. $K_i(\varphi \to \psi) \to K_i\varphi \to K_i\psi$ for $i = 1, ..., m$.
2. $[v_i](\varphi \to \psi) \to [v_i]\varphi \to [v_i]\psi$ for $i = 1, ..., m$.
3. $\Box(\varphi \to \psi) \to \Box\varphi \to \Box\psi$ for $i = 1, ..., m$.
4. $B_i\varphi \leftrightarrow \neg K_i \neg\varphi$.
5. $< v_i > \varphi \leftrightarrow \neg[v_i]\neg\varphi$.
6. $\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$.
7. $\Box\varphi \to (v_i)\varphi$ for some event v_i for $i = 1, ..., m$.
8. $(v_i)\varphi \to < v_i > \varphi$ for some event v_i for $i = 1, ..., m$.
9. $< v_i > \varphi \to (v_j)\varphi$ for some event v_j for $i, j = 1, ..., m$.
10. $K_i[v_i]\varphi \to [v_i]K_i\varphi$ for $i = 1, ..., m$.
11. $K_i(v_i)\varphi \to (v_i)K_i\varphi$ for $i = 1, ..., m$.
12. Modus Ponens: From j and $\varphi \to \psi$ derive ψ.

13. Generalization of $\Box$ (always on future): From $\models \varphi$ derive $\Box\varphi$.

14. Generalization of $[v_i]$: From $\models \varphi$ derive $[v_i]\varphi$.

5 Example Application

In order to illustrate the applicability of our logic, we present the Propagation Information with Feedback algorithm (PIF) for a three agents system. The precedence graph for a run of PIF is shown in figure 5.

Following the run r_1 in figure 5, we have that after agent a_1 sends message φ to a_2 and a_3 (what he does on event e_1) he knows that a_2 and a_3 will know φ in the future: $K_1(< e_1 > (K_2\varphi \wedge K_3\varphi))$. Considering agents a_2 and a_3 only begin their computations after receiving the initial message from a_1, after the occurrence of e_1 there will be state change only when a_2 or a_3 receives that message from a_1. So, after the occurrence of the next event, one among a_2 or a_3 will know φ. After the occurrence of e_1 holds: $\Box(K_2\varphi \vee K_3\varphi)$, $\diamond K_2\varphi$, $\diamond K_3\varphi$.

On receiving the message from a_1, a_2 learns that a_1 knows φ, and a_3 goes through the same: $[e_2]K_2K_1\varphi$, $[e_3]K_3K_2\varphi$. And to agents a_2 and a_3 after e_5 and e_4 respectively: $[e_5]K_2K_3\varphi$, $[e_4]K_3K_1\varphi$.

After e_2, agent a_2 forwards the message to a_3, and so becomes conscious that a_3 will know φ in the future. The same happens when a_3 forwards the message to a_1 after e_3: $K_2(< e_2 > K_3\varphi)$, $K_3(< e_3 > K_1\varphi)$. Analogously, well have: $[e_6]K_1K_3\varphi$, $[e_7]K_1K_2\varphi$, $[e_6]K_1K_3K_1\varphi$, $[e_7]K_1K_2K_1\varphi$.

6 Conclusions

The main contribution of the paper is the presentation of a logic capable of reasoning about knowledge and time in asynchronous distributed systems. The handling of time is done using an event-based formalism in such way that the language can model the evolution of knowledge in completely asynchronous distributed systems. New event-based temporal operators were defined, providing a greater temporal expressivity to the language, even considering time under the partial ordering of events imposed by the ADS model in [2]. Using a Kripke structure to support knowledge turns the Event-based Knowledge Logics compatible with all kinds of knowledge operators based on the possible worlds concept defined to asynchronous models. As future works, we would like to present an axiomatic system to the language, implement a model check and a theorem prover. Another direction is to define operators to represent group knowledge, specially an operator for common knowledge like the concurrent common knowledge operator shown in [4].

Acknowledgments

The authors have received partial support from the Brazilian Research Agencies CNPq, CAPES and FAPERJ.

References

[1] FAGIN, R., HALPERN, J. Y., MOSES, Y. and others: Reasoning About Knowledge. MIT Press. Cambridge, Massachussetts (1995)

[2] LAMPORT, L.: Time, Clocks, and the ordering of events in a distributed system. Comm. of the ACM, Vol. 206(21), (1978) 558–565

[3] LEHMANN, D. J.: Knowledge, Common Knowledge and related puzzles (Extended Summary). In: Proceedings of the Third Annual ACM Symposium on Principles of Distributed Computing, Vancouver, Canada, (1984) 62–67

[4] PANANGADEN, P., TYLER, K.: Concurrent Common Knowledge: Defining Agreement for Asynchronous Systems. Distributed Computing, Vol. 6(2),(1992) 73–93