Testing of Synchronization Conditions for Distributed Real-time Applications

Ajay D. Kshemkalyani

Dept. of Electrical and Computer Engineering and Computer Science
P. O. Box 210030, University of Cincinnati, Cincinnati, OH 45221-0030, USA
Email: ajayk@ececs.uc.edu

Abstract. A set of synchronization relations between distributed nonatomic events was recently proposed to provide real-time applications with a fine level of discrimination in the specification of causality relations and synchronization conditions. For a pair of distributed nonatomic events X and Y, the evaluation of the synchronization relations requires $|N_X| \times |N_Y|$ integer comparisons, where $|N_X|$ and $|N_Y|$, respectively, are the number of nodes on which the two nonatomic events X and Y occur. In this paper, we show that this polynomial complexity of evaluation can by simplified using properties of partial orders to a linear complexity. Specifically, we show that most relations can be evaluated in $min(|N_X|, |N_Y|)$ integer comparisons, some in $|N_X|$ integer comparisons, and the others in $|N_Y|$ integer comparisons. These linear time evaluation conditions enable the real-time applications to detect the relations efficiently.

Keywords: Time, Synchronization, Distributed system, Efficient measures.

1 Introduction

Several distributed applications are characterized by real-time constraints on response times. High-level actions in such distributed real-time application executions [11, 12] are realistically modeled by nonatomic poset, i.e., nonlinear, events (where at least some of the component atomic events of a nonatomic event occur at more than a single point in space concurrently), for example, in industrial process control, distributed multimedia support, coordination in mobile computing, avionics, terrestrial, undersea and aerial navigation, planning, robotics, and virtual reality. It is important to provide these and emerging sophisticated real-time applications a fine level of discrimination in the specification of various synchronization/causality relations between nonatomic poset events. In addition, [20] stressed the theoretical and practical importance of the need for such relations. Most of the existing literature [1, 2, 4, 5, 6, 7, 10, 14, 16, 17, 18, 19, 20] does not address this issue. A set of causality relations between nonatomic poset events was proposed in [8, 11, 12] to specify and reason with a fine-grained specification of causality. This set of causality relations [8, 11, 12] extended the hierarchy of the relations in [9, 15]. Specific use of the proposed relations in distributed mutual exclusion and distributed predicate specification in the context of a real-time air defence control system was also demonstrated in [11]. An axiom system on the proposed relations was given in [13]. The objective of this paper is to derive efficient test conditions for the relations in [11, 12].

We adopt the following poset event structure model as in [4, 9, 10, 11, 14, 15, 16, 20]. Consider a poset $(E, \prec)$ where $\prec$ is an irreflexive partial ordering that represents the causality relation. $(E, \prec)$ represents points in space-time which are the most primitive atomic events related by the causality relation. Elements of E are partitioned into local executions at a coordinate in the space dimensions. In a distributed system, E represents a set of events and is discrete. Each local execution E_i is a linearly ordered set of events in partition i. An event e in partition i is denoted e_i. For a distributed computing system, points in the space dimension correspond to the set of processes (also termed *nodes*), and E_i is the set of events executed by process i. Causality between events at different nodes is imposed by message communication. We also assume there are a finite number of nodes i, and each E_i has a dummy initial event ($\perp_i$) and a dummy final event ($\top_i$). Let $E^{\perp}$ and $E^{\top}$ denote the sets of initial events and final events, respectively. We assume that $\forall \perp_i \forall \top_j \; \forall e \in (E \setminus E^{\perp} \setminus E^{\top})$, $\perp_i \prec e \prec \top_j$.

Nonatomic nonlinear events are defined as follows. Let $\mathcal{E}$ denote the power set of E. Let $\mathcal{A}$ $(\neq \emptyset) \subseteq (\mathcal{E} - \emptyset)$. Thus, there is an implicit one-many mapping from $\mathcal{A}$ to E. Each element A of $\mathcal{A}$ is a non-empty subset of E, and is termed an *interval* or a *nonatomic event*. It follows that if $A \bigcap E_i \neq \emptyset$, then $(A \bigcap E_i)$ has a least and a greatest event. Typically, $\mathcal{A}$ is the set of all the sets that represent a higher level grouping of the events of E that is of interest to an application. An event A of interest to an application will usually not contain any dummy events. We denote $A \bigcap E_i$ as A_i.

We define the *node set* of a nonatomic event to be the set of nodes at which its component atomic events occur.

Definition 1. N_A, the node set of event A, is $\{i \mid E_i \bigcap A \not\subseteq \{\perp_i, \top_i\}\}$.

The relations proposed in [9] formed an exhaustive set of causality relations to express all possible interactions between a pair of linear intervals. The relations $R1$ - $R4$ and $R1'$ - $R4'$ from [9] are expressed in terms of the quantifiers over X and Y in Table 1. For $R1'$, $R2'$, $R3'$, and $R4'$, the order of quantifiers was reversed from the order in $R1$, $R2$, $R3$, and $R4$, respectively. Observe that the relations $R2'$ and $R3'$ are different from relations $R2$ and $R3$, respectively, when applied to posets.

Relation R	Expression for $R(X, Y)$	Evaluation condition using relation $\ll$ between cuts
$R1$ $R1'$	$\forall x \in X \forall y \in Y, x \prec y$ $\forall y \in Y \forall x \in X, x \prec y$	$\prod_{x \in X}[\cap_{\Downarrow} Y \not\ll x\uparrow]$ $= \prod_{y \in Y}[\downarrow y \not\ll \cup_{\Uparrow} X]$
$R2$	$\forall x \in X \exists y \in Y, x \prec y$	$\prod_{x \in X}[\cup_{\Downarrow} Y \not\ll x\uparrow]$
$R2'$	$\exists y \in Y \forall x \in X, x \prec y$	$\cup_{\Downarrow} Y \not\ll \cup_{\Uparrow} X$
$R3$	$\exists x \in X \forall y \in Y, x \prec y$	$\cap_{\Downarrow} Y \not\ll \cap_{\Uparrow} X$
$R3'$	$\forall y \in Y \exists x \in X, x \prec y$	$\prod_{y \in Y}[\downarrow y \not\ll \cap_{\Uparrow} X]$
$R4$ $R4'$	$\exists x \in X \exists y \in Y, x \prec y$ $\exists y \in Y \exists x \in X, x \prec y$	$\cup_{\Downarrow} Y \not\ll \cap_{\Uparrow} X$

Table 1. Relations in [9] are given in the first two columns. The third column (explained later) gives the evaluation conditions derived in this paper. $\prod$ is the product or conjunction operator.

When the relations of [9] are applied to a pair of poset intervals, the hierarchy they form is incomplete. In [8, 11, 12], we formulated causality relations between a pair of nonatomic poset intervals along the lines of [9] by extending these results to nonatomic poset events. The relations form an “exhaustive” set of causality relations between nonatomic poset events using first-order predicate logic and only the relation $\prec$ between atomic events, and fill in the partial hierarchy of causality relations between nonatomic poset events, formed by relations in [9, 15].

The causality relations between a pair of nonatomic poset events were formulated in [8, 11, 12] using the notion of *proxies*. For each nonatomic poset event X, two proxies L_X and U_X to represent its beginning and end, respectively, were defined using Definition 2 or 3. These proxies were the equivalents of the beginning and end instants of a nonatomic linear event [1, 2, 5].

Definition 2. • $L_X = \{e_i \in X | \forall e'_i \in X, e_i \preceq e'_i\}$ • $U_X = \{e_i \in X | \forall e'_i \in X, e_i \succeq e'_i\}$

Definition 3. • $L_X = \{e \in X | \forall e' \in X, e \preceq e'\}$ • $U_X = \{e \in X | \forall e' \in X, e \succeq e'\}$

Any of the above or a similar definition of proxies is consistently used, depending on context and application. We denote a proxy of X as $\hat{X}$. Figure 1 depicts the proxies of X and Y and serves as a visual aid for the following discussion.

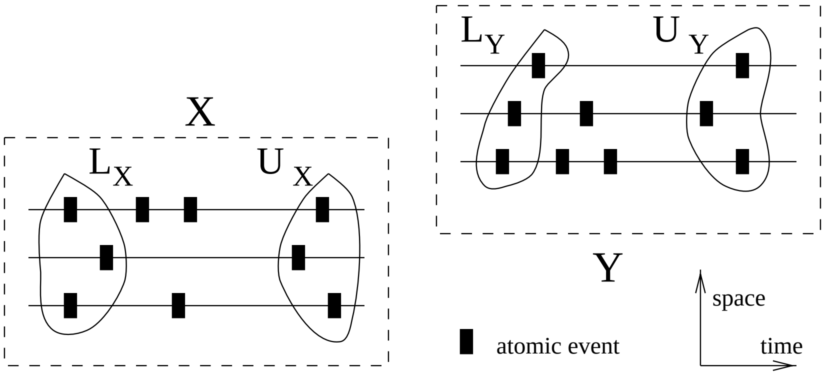

Fig. 1. Poset events X and Y and their proxies.

The causality relations in [8, 11, 12] were defined using two aspects of specifying the relations. Causality relations between poset intervals can be specified between the proxies of X and Y. As there is a choice of two proxies of X and choice of two proxies of Y, there are four combinations between the proxies $\hat{X}$ and $\hat{Y}$. The eight causality relations in Table 1 can be specified for each combination, thus yielding 32 relations between X and Y. We denote the set of these causality relations as $\mathcal{R}$. From the construction of $\mathcal{R}$, it follows that for nonatomic poset events X and Y, there is a 1-1 equivalence between any $r(X, Y)$, for $r \in \mathcal{R}$, and $R(\hat{X}, \hat{Y})$, for some R in Table 1 and some $\hat{X}$ and some $\hat{Y}$.

Notation: We use the notation $\hat{X}$ to specifically distinguish a subset of E that acts as a proxy for another subset X of E. Otherwise, when the distinction is not important, the notation X refers to any subset of E, which can also be a proxy of another set.

Objectives. Given a trace of a distributed execution, the application identifies pertinent nonatomic events and needs to know what relations are satisfied between pairs of such events. Implicit in the use of these relations is the need to detect whether some specific relation holds between a given pair of nonatomic events (see Problem 4).

Problem 4. Given a recorded trace of a distributed computation $(E, \prec)$ and a set of nonatomic events $\mathcal{A}$, then for every pair of nonatomic poset events X and Y, where $X, Y \in \mathcal{A}$, **efficiently** determine (i) if a specific relation $r(X,Y)$ holds, for $r \in \mathcal{R}$, and, (ii) all the relations $r(X,Y)$ that hold, for $r \in \mathcal{R}$.

Problem 1 can be answered by testing for the appropriate causality relation(s) of Table 1 on $\hat{X}$ and $\hat{Y}$. Observe from the second column of Table 1 that each relation $r(X,Y)$, for $r \in \mathcal{R}$, (which corresponds exactly to some $R(\hat{X}, \hat{Y})$, for some R in Table 1) can be evaluated with $|N_X| \times |N_Y|$ checks [1] for causality. This is significantly better than $|X| \times |Y|$ checks for causality that would be needed without the use of proxies in the definitions of causality. However, this evaluation has a polynomial computational complexity ($|N_X| \times |N_Y|$ checks for causality). Our objective is to simplify the test for the relations. In this paper, we show that the evaluation of the relations can by further simplified using properties of partial orders.

Recall that for nonatomic poset events X and Y, there is a 1-1 equivalence between any $r(X,Y)$, for $r \in \mathcal{R}$, and $R(\hat{X}, \hat{Y})$, for some R in Table 1 and some $\hat{X}$ and some $\hat{Y}$. But $\hat{X}$ and $\hat{Y}$ are themselves nonatomic poset events like X and Y – the only difference is that for any node i, $|\hat{X}_i| \leq 1$ and $|\hat{Y}_i| \leq 1$, whereas $|X_i|$ and $|Y_i|$ are bounded only by $|E_i|$. We show that the evaluation methodology and complexity of $R(X,Y)$ is independent of the size of $|X_i|$ and $|Y_i|$. Hence, we derive the evaluation methodology for $R(X,Y)$, where R belongs to Table 1. Then, using a suitable quantification of X and Y in these results to represent the various proxies $\hat{X}$ and $\hat{Y}$, we obtain the evaluation methodology for each of the 32 relations in $\mathcal{R}$. The simplified evaluation conditions we derive have only a linear computational complexity for each relation.

The main result in the paper (Theorem 20) therefore shows that relations $R1$, $R1'$, $R2'$, $R3$, $R4$, and $R4'$ can be evaluated in $min(|N_X|, |N_Y|)$ integer comparisons, relation $R2$ in $|N_X|$ integer comparisons, and relation $R3'$ in $|N_Y|$ integer comparisons.

Sections 2.1 and 2.2 introduce execution prefixes associated with nonatomic events, and the $\ll$ relation between such prefixes. Section 2.2 informally shows the equivalence between $R(X,Y)$, for R in Table 1, and the $\ll$ relation between certain prefixes associated with X and Y. Section 2.3 develops timestamps of execution prefixes associated with nonatomic events, and Section 2.4 develops an efficient test for the $\ll$ relation on such prefixes. Combining the results of Sections 2.2 and 2.4, Section 2.5 determines the exact complexity of testing for $R(X,Y)$, for R in Table 1. Section 3 concludes. Proofs of theorems and lemmas are given in [8].

2 Efficient Evaluation of Causality Relations

2.1 Cuts of an Execution

Let P be the set of all process/node partitions. An execution prefix or a *cut* is the union of a downward-closed subset of each E_i, for every node $i \in P$.

[1] We use the terms $|N_X|$ and $|N_Y|$ which are upper bounds on $|N_{\hat{X}}|$ and $|N_{\hat{Y}}|$, respectively.

Definition 5. A cut C is the union of a downward-closed subset of each E_i in $(E, \prec)$, where $E = \bigcup_{\forall i \in P} E_i$.

$$C \equiv C \subseteq E \bigwedge E^{\perp} \subseteq C \bigwedge e_i \in C \Longrightarrow (\forall e'_i,\ e'_i \prec e_i \Longrightarrow e'_i \in C)$$

A cut has a well-defined upper and lower bound at each node in its node set. Next, we define $S(C)$ to be the set of latest events at each node in cut C. $S(C)$ denotes the "surface" of cut C and is the same as the proxy $U(C)$ if $U(C)$ is defined by Definition 2.

Definition 6. • $S(C) = \{e_i \in C \mid \forall e'_i \in C, e_i \succeq e'_i\}$

Given a cut C, C_i (or $[S(C)]_i$) is a subset of C (or $S(C)$) that contains elements in partition i. Thus, C (or $S(C)$) is projected over partition i.

Comparison of Cuts. It is known from lattice theory that the set of all cuts, denoted $\mathcal{C}$, forms a lattice ordered by $\subset$. We introduce a new relation $\ll$ over the set of cuts. $\ll(C, C')$ signifies that cut C is a proper subset of cut C' and moreover, C_i is a proper subset of C'_i. This relation is useful to derive simplified evaluation conditions for the relations between nonatomic poset events.

Definition 7. We express the relation $\ll(C, C')$ in different forms, each of which will be used subsequently.

1. $\ll(C, C')$ iff $(\forall z \in (S(C) \setminus E^{\perp}), z \notin S(C') \bigwedge z \in C') \bigwedge C' \neq E^{\perp}$.
2. $\not\ll(C, C')$ iff $(\exists z \in (S(C) \setminus E^{\perp}), z \in S(C') \bigvee z \notin C') \bigvee C' = E^{\perp}$.
3. $\ll(C, C')$ iff $(\forall z \in (S(C') \setminus E^{\perp}), z \notin C) \bigwedge C' \neq E^{\perp} \bigwedge N_C \subseteq N_{C'}$.
4. $\not\ll(C, C')$ iff $(\exists z \in (S(C') \setminus E^{\perp}), z \in C) \bigvee C' = E^{\perp} \bigvee N_C \not\subseteq N_{C'}$.

All the four forms of the definition can be seen to be equivalent. The terms $C' \neq E^{\perp}$ and $C' = E^{\perp}$ are required to make the definitions robust in certain cases where $C' = E^{\perp}$. The forms in Definition 7.2 and Definition 7.4 express the condition for $\not\ll(C, C')$ which we will use subsequently as follows. The significance of $\not\ll$ is that if $\not\ll(C, C')$, then some event in $S(C)$ (equals or) happens causally after some event in $S(C')$. If we can choose C and C' appropriately to correspond to X and Y, for any $R(X, Y)$, for $R \in$ Table 1, then we have a reexpression for the relation R. Then the evaluation of $R(X, Y)$ which requires at least $|N_X| \times |N_Y|$ checks for causality reduces to the evaluation of $\not\ll(C, C')$ which takes $|P|$ evaluations in the general case. But C and C' are not arbitrary cuts; rather, they are the cuts identified by X and Y and are structured based on the membership of X and Y. Therefore, the number of checks for causality can be further reduced.

2.2 Past and Future Cuts of a Poset Event

For atomic event e, there are two special cuts $\downarrow e$ and $e\uparrow$. $\downarrow e$ is the maximal set of events that happen before or equal e. $\downarrow e$ denotes the causal past (CP) of e. $e\uparrow$ is the union of a downward-closed subset of events at each node, such that the maximum element of the downward-closed subset at any node i is the earliest event at i for which e happens before or equals the event. $e\uparrow$ is the complement of the causal future (CCF) of e and denotes the execution prefix upto and including the beginning of the causal future of e at each node.

Definition 8. [CP:] $\downarrow e \equiv \{e' \mid e' \preceq e\}$

Definition 9. [CCF:] $e\uparrow \equiv \{e' \mid e' \not\preceq e\} \bigcup \{e_i, i \in P \mid e_i \succeq e \bigwedge (\forall e'_i,\ e'_i \prec e_i \Longrightarrow e'_i \not\preceq e)\}$

The cuts $\downarrow e$ and $e\uparrow$ have the property that cut $\downarrow y$ has a unique maximal event and cut $x\uparrow$ has a unique minimal event. Also, $\downarrow e$ is downward-closed in $(E, \prec)$; $e\uparrow$ is not.

Given a poset event, we define certain cuts that represent the past and the future of the execution associated with the poset event; each cut has a different significance.

Label	Definition	Timestamp, derived from Defn. 15 and Lemma 16
$C1(X)$ or $\cap_{\Downarrow} X$	$\bigcap_{\forall x \in X} \{\downarrow x\}$	$T(\cap_{\Downarrow} X) \equiv \forall i \in P, T(\cap_{\Downarrow} X)[i] = min_{\forall x \in X}(T(\downarrow x)[i])$
$C2(X)$ or $\cup_{\Downarrow} X$	$\bigcup_{\forall x \in X} \{\downarrow x\}$	$T(\cup_{\Downarrow} X) \equiv \forall i \in P, T(\cup_{\Downarrow} X)[i] = max_{\forall x \in X}(T(\downarrow x)[i])$
$C3(X)$ or $\cap_{\Uparrow} X$	$\bigcap_{\forall x \in X} \{x\uparrow\}$	$T(\cap_{\Uparrow} X) \equiv \forall i \in P, T(\cap_{\Uparrow} X)[i] = min_{\forall x \in X}(T(x\uparrow)[i])$
$C4(X)$ or $\cup_{\Uparrow} X$	$\bigcup_{\forall x \in X} \{x\uparrow\}$	$T(\cup_{\Uparrow} X) \equiv \forall i \in P, T(\cup_{\Uparrow} X)[i] = max_{\forall x \in X}(T(x\uparrow)[i])$

Table 2. Definitions of special sets of poset X. These sets are shown to be cuts. Timestamps of the cuts are given in the third column.

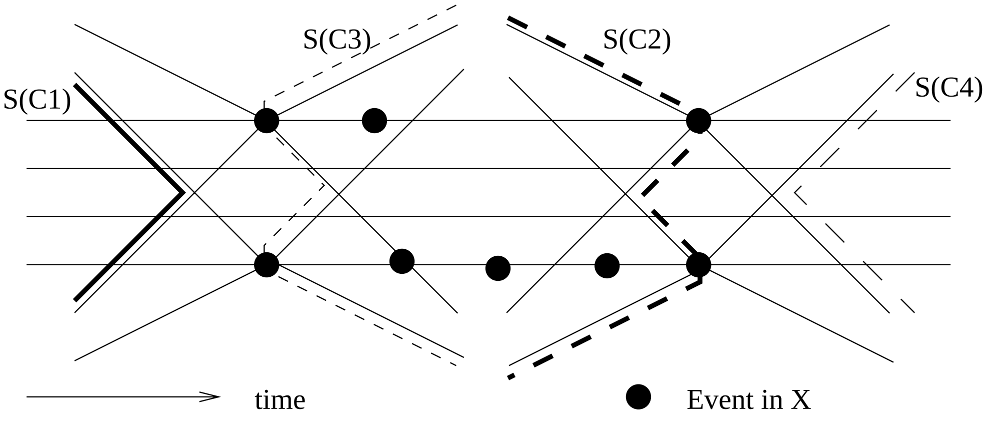

Fig. 2. Cuts of poset X which contains 8 atomic events.

Definition 10. The second column of Table 2 defines certain sets associated with poset event X.

Lemma 11. *The sets defined in Definition 10 are cuts.*

Figure 2 illustrates the cuts C1–C4 defined in Table 2 for a poset X containing eight elements that are marked by shaded circles. The four horizontal lines are the time lines of four nodes. There is a computation event (not shown) at the intersection of each horizontal time line and a cut C1–C4. The surface of each cut is marked and labeled as follows: $S(C1)$, $S(C2)$, $S(C3)$, and $S(C4)$ are marked by a thick line, a thick dashed line, a dashed line, and a far-spaced dashed line, respectively.

The cuts $\cap_{\Downarrow} X$ and $\cup_{\Downarrow} X$ which are determined by the set of cuts $\{\downarrow x \mid x \in X\}$ condense the causality information in each cut in the set, i.e., information about the past of the execution associated with events in X. The cuts $\cap_{\Uparrow} X$ and $\cup_{\Uparrow} X$ which are determined by the set of cuts $\{x\uparrow \mid x \in X\}$ condense the causality information in each cut in the set, i.e., information about the future of the execution associated with events in X. Observe that $\cap_{\Downarrow} X$ and $\cup_{\Downarrow} X$ are downward-closed subsets of $(E, \prec)$; $\cap_{\Uparrow} X$ and $\cup_{\Uparrow} X$ are not.

Lemma 12. *The members of a poset are related to the cuts associated with the poset, defined in Definition 10, as follows.*

12.1 $\forall e' \in S(\cap_{\Downarrow} X) \forall x \in X,\ e' \preceq x$
12.2 $\forall e' \in S(\cup_{\Downarrow} X) \exists x \in X,\ e' \preceq x$
12.3 $\forall e' \in S(\cap_{\Uparrow} X) \exists x \in X,\ x \preceq e'$
12.4 $\forall e' \in S(\cup_{\Uparrow} X) \forall x \in X,\ x \preceq e'$

The cuts of a nonatomic poset event defined in Definition 10 represent various execution prefixes associated with the nonatomic event. Cuts C1(X) and C2(X) are about the past of the nonatomic event and cuts C3(X) and C4(X) are about the future of the nonatomic event. The significance of these cuts is discussed and expressed in knowledge-theoretic terminology next [3]. We will use notation Φ_X and Φ_{cut} to represent knowledge about nonatomic event X and cut cut, respectively. $K_x(\Phi)$ is a predicate that is true if event x has knowledge of Φ. Ψ^x represents the knowledge available at event x.

1. $\cap_{\Downarrow} X$ is the maximum set of events that causally precede every $x \in X$. It represents the maximum execution prefix about which all events in X have knowledge.
   In knowledge-theoretic terms, $\forall x {\in} X,\ K_x(\Phi_{\cap_{\Downarrow} X}) = true$. Also, $\forall x {\in} X,\ \Phi_{\cap_{\Downarrow} X} \subseteq \Psi^x$.
2. $\cup_{\Downarrow} X$ is the maximum set of events such that each event causally precedes some $x \in X$. It represents the maximum execution prefix about which only all the events in X collectively have knowledge, but no one event in X may have complete knowledge.
   In knowledge-theoretic terms, $\bigcup_{\forall x \in X} (\Psi^x) = \Phi_{\cup_{\Downarrow} X}$. Also, $\forall e_i \in S(\cup_{\Downarrow} X)\ \exists x \in X, \Psi^x \supseteq \Psi^{e_i}$.
3. $\cap_{\Uparrow} X$ is a cut such that $S(\cap_{\Uparrow} X)$ is the set of earliest events on each node that are causally preceded by some $x \in X$. It represents the minimum execution prefix such that all the maximum events of this prefix are preceded by at least one event in X.
   In knowledge-theoretic terms, $\forall e_i \in S(\cap_{\Uparrow} X)\ \exists x \in X,\ K_{e_i}(\Phi_x) = true$. Also, $\forall e_i \in S(\cap_{\Uparrow} X)\ \exists x \in X, \Psi^x \subseteq \Psi^{e_i}$.
4. $\cup_{\Uparrow} X$ is a cut such that $S(\cup_{\Uparrow} X)$ is the set of earliest events on each node that are causally preceded by every $x \in X$. It represents the minimum execution prefix such that all the maximum events of this prefix are causally preceded by all the events in X.
   In knowledge-theoretic terms, $\forall e_i {\in} S(\cup_{\Uparrow} X),\ K_{e_i}(\Phi_X) = true$. Also, $\forall e_i \in S(\cup_{\Uparrow} X)$ $\forall x \in X, \Psi^x \subseteq \Psi^{e_i}$.

Key Idea 1: The cuts $\cap_{\Downarrow}X$, $\cup_{\Downarrow}X$, $\cap_{\Uparrow}X$, and $\cup_{\Uparrow}X$ aggregate the causality information about all x in a nonatomic event X in a condensed form, as described above. Once identified at a one-time cost, these cuts can be reused at a low cost to evaluate causality relations with respect to all other nonatomic events.

We now informally show the equivalence of (i) the relations $R(X, Y)$, for R in the second column of Table 1, and (ii) the relation $\not\ll$ on appropriately identified cuts $C1$, $C2$, $C3$, and $C4$ associated with X and Y as given in the third column of Table 1, using Lemma 12 and the knowledge-theoretic analysis of the cuts. (See [8] for a formal proof.) Note that if $\not\ll(C, C')$, then some event in $S(C')$ happens before (or equals) some event in $S(C)$. But, in the following discussion, we assume that if $\not\ll(C, C')$, then some event in $S(C')$ happens before some event in $S(C)$.

$R1(X, Y)$**:** This relation holds iff $\forall x \in X$, $\not\ll(\cap_{\Downarrow}Y, x\!\uparrow)$, i.e., $\forall x \in X$, some event in $S(\cap_{\Downarrow}Y)$ happens causally after some event in $S(x\!\uparrow)$, implying by the use of a transitive argument and Lemma 12.1 that for all events x in X, all events in Y happen causally after x.

$R1'(X, Y)$**:** This relation holds iff $\forall y \in Y$, $\not\ll(\downarrow\! y, \cup_{\Uparrow}X)$, i.e., $\forall y \in Y$, some event in $S(\downarrow\! y)$ happens causally after some event in $S(\cup_{\Uparrow}X)$, implying by the use of a transitive argument and Lemma 12.4 that for all events y in Y, y happens causally after all events in X.

$R2(X, Y)$**:** This relation holds iff $\forall x \in X$, $\not\ll(\cup_{\Downarrow}Y, x\!\uparrow)$, i.e., $\forall x \in X$, some event in $S(\cup_{\Downarrow}Y)$ happens causally after some event in $S(x\!\uparrow)$, implying by the use of a transitive argument and Lemma 12.2 that for all events x in X, some event in Y happens causally after x.

$R2'(X, Y)$**:** This relation holds iff $\not\ll(\cup_{\Downarrow}Y, \cup_{\Uparrow}X)$, i.e., some event in $S(\cup_{\Downarrow}Y)$ happens causally after some event in $S(\cup_{\Uparrow}X)$, implying by the use of a transitive argument and Lemmas 12.2 and 12.4 that some event in Y happens causally after all the events in X.

$R3(X, Y)$**:** This relation holds iff $\not\ll(\cap_{\Downarrow}Y, \cap_{\Uparrow}X)$, i.e., some event in $S(\cap_{\Downarrow}Y)$ happens causally after some event in $S(\cap_{\Uparrow}X)$, implying by the use of a transitive argument and Lemmas 12.1 and 12.3 that for some event in X, all the events in Y happen causally after that event in X.

$R3'(X, Y)$**:** This relation holds iff $\forall y \in Y$, $\not\ll(\downarrow\! y, \cap_{\Uparrow}X)$, i.e., $\forall y \in Y$, some event in $S(\downarrow\! y)$ happens causally after some event in $S(\cap_{\Uparrow}X)$, implying by the use of a transitive argument and Lemma 12.3 that for all events y in Y, y happens causally after some event in X.

$R4(X, Y)$**,** $R4'(X, Y)$**:** This relation holds iff $\not\ll(\cup_{\Downarrow}Y, \cap_{\Uparrow}X)$, i.e., some event in $S(\cup_{\Downarrow}Y)$ happens causally after some event in $S(\cap_{\Uparrow}X)$, implying by the use of a transitive argument and Lemmas 12.2 and 12.3 that some event in Y happens causally after some event in X.

2.3 Timestamps

Timestamps of Atomic Events. Each atomic event is assigned a timestamp which is the clock value when the event occurs. Clocks are such that the timestamps of events have the following property: $e \prec e'$ iff $T(e) < T(e')$ [4, 16]. The canonical vector clocks in [4, 16] have the above property. Each primitive atomic event e is assigned a timestamp $T(e)$ that is a vector of size $|P|$, where P is the set of all process/node partitions. This is the minimum size of a clock/timestamp that is required to capture the above property of timestamps. Assuming that the identifier of a process/node i is i itself, $T(e)$ is defined as follows.

Definition 13. $T(e) \equiv \forall i \in P, T(e)[i] = |\{e_i \mid e_i \preceq e\}|$, i.e., $T(e)[i]$ is the number of events on node i that causally precede or equal e.

Let $\mathcal{T}$ be the set $\{T(e) | e \in E\}$. Note that $(E, \prec)$ is isomorphic to $(\mathcal{T}, <)$.

Analogous to the timestamp $T(e)$, the reverse timestamp $T^R(e)$ of an event indicates the number of events in the future that are causally affected by the current event.

Definition 14. $T^R(e) \equiv \forall i \in P, T^R(e)[i] = |\{e_i \mid e_i \succeq e\}|$, i.e., $T^R(e)[i]$ is the number of events on node i that causally happen after or equal e.

Observe that once the timestamp structure is established for the entire computation, the "reverse" timestamp structure can also be established.

Given two distinct atomic events e_j and e'_k, the causality between them can be tested as follows: $e_j \prec e'_k \ \ iff \ \ T(e_j)[j] < T(e'_k)[j]$.

Timestamps of Cuts and Nonatomic Events. For a cut C, we define its timestamp $T(C)$ such that the ith component of the timestamp is the maximum of the ith components of the timestamps of all the events in C that occur at node i.

Definition 15. $T(C) \equiv \forall i \in P, T(C)[i] = max_{\forall x_i \in C}(T(x)[i])$

Observe that $T(C)[i]$ is the same as $T([S(C)]_i)[i]$, i.e., $T(C)[i]$ is the ith component of the timestamp of the latest event in C that occurs at i.

Lemma 16. *The timestamp of a cut composed by the union or intersection of other cuts is as follows.*

- *If* $C \equiv \bigcap_{s=1,k} C^s$ *then* $T(C) \equiv \forall i \in P, \ T(C)[i] = min_{s=1,k}(T(C^s)[i])$
- *If* $C \equiv \bigcup_{s=1,k} C^s$ *then* $T(C) \equiv \forall i \in P, \ T(C)[i] = max_{s=1,k}(T(C^s)[i])$

Corollary 17. *The timestamps of the cuts of a poset defined in the second column of Table 2 are given in the third column of the table.*

Causality between nonatomic poset events X and Y is determined as follows. Compare the timestamp of an appropriately chosen cut associated with X with the timestamp of an appropriately chosen cut associated with Y to test for the $\ll$ relation between the two cuts. Then formally show that this test (possibly multiple such tests) is equivalent to the test for causality (Section 2.5).

From Defns. 8 and 13, observe that $T(\downarrow x)$, the timestamp of cut $\downarrow x$ associated with any event x is simply $T(x)$. From Defns. 9 and 14, observe that $T(x\uparrow)$, the timestamp of cut $x\uparrow$ associated with any event x is as follows: $T(x\uparrow)[i] = |E_i| - T^R(x)[i] - 1$. (This expression accounts for the two dummy events in E_i.) Using timestamps of cuts $\downarrow x$ and $x\uparrow$, the overhead of computing timestamps of the cuts given in Table 2 for each

X is as follows. The i^{th} component of the timestamp of each of $C1(X)$ and $C2(X)$ is a min and max function, respectively, of $\{T([S(\downarrow x)]_i)[i] \mid x \in X\}$. Similarly, the i^{th} component of the timestamp of each of $C3(X)$ and $C4(X)$ is a min and max function, respectively, of $\{T([S(x\uparrow)]_i)[i] \mid x \in X\}$. Observe that for $C1(X)$ and $C3(X)$, it suffices to consider only the least element in $X \bigcap E_i$, for each $i \in N_X$. Similarly, observe that for $C2(X)$ and $C4(X)$, it suffices to consider only the latest element in $X \bigcap E_i$, for each $i \in N_X$. Hence, the i^{th} component of the timestamp of each of $C1(X)$, $C2(X)$, $C3(X)$, and $C4(X)$ is a min or max function over the i^{th} components of $|N_X|$ timestamps, which has a $|N_X|$ computational complexity. For $|P|$ components of the timestamp, the computational complexity is $|N_X| \times |P|$. Fortunately, we show in Theorem 19 (using Key Idea 2) that all the $|P|$ components of the timestamps of the cuts are not required for computing the $\ll$ relation between the cuts. Rather, for event X, only the $|N_X|$ components for the nodes in N_X are relevant, and hence, only these need to be computed. So the computational complexity of computing the timestamp of a cut $C1(X)$, $C2(X)$, $C3(X)$, or $C4(X)$ is $|N_X|^2$. Observe that this computation of the timestamps of the above cuts is a one-time cost (analyzed in [8] and shown to be negligible). Once computed, the timestamp of a cut associated with X can be reused in the evaluation of the relation $\ll$ between this cut and cuts associated with multiple other nonatomic events.

2.4 Efficient Evaluation of $\ll$ between Past and Future Cuts of Posets

Notation: We use $\downarrow X$ and $X\uparrow$ to denote cuts about the past and future associated with any nonempty subset of E, respectively.

Definition 18. For any nonatomic poset event X,
- $\downarrow X$ denotes either $C1(X)$ or $C2(X)$.
- $X\uparrow$ denotes either $C3(X)$ or $C4(X)$.

We showed at the end of Section 2.2 that each of the 8 relations $R(X, Y)$ in Table 1 can be expressed as $\not\ll (C, C')$, where C and C' are appropriately identified in that section. Thus, we have a reexpression for the relation R in terms of $\ll$. Then the evaluation of $R(X, Y)$ which requires at least $|N_X| \times |N_Y|$ checks for causality reduces to the evaluation of $\not\ll(C, C')$ which takes $|P|$ evaluations in the general case. But C and C' are not arbitrary cuts; rather, they are the cuts identified by X and Y and are structured based on the membership of X and Y. Specifically, C and C' are the cuts $\downarrow Y$ and $X\uparrow$ which are determined by the sets of cuts $\{\downarrow y \mid y \in Y\}$ and $\{x\uparrow \mid x \in X\}$, respectively, which have the property that each cut $\downarrow y$ has a unique maximal event and each cut $x\uparrow$ has a unique minimal event. This property suggests that sufficient causal information about X and Y is condensed into the N_X and N_Y components, respectively, of each of the above cuts and their surfaces, and leads to the following idea.

Key Idea 2: If $\ll(\downarrow Y, X\uparrow)$ is violated, then some event in $S(\downarrow Y)$ equals or happens causally after some event in $S(X\uparrow)$. This violation must occur at a node in N_X because the events $[S(X\uparrow)]_{N_X}$ are the earliest possible events among events in $S(X\uparrow)$, in terms of causality. Using analogous reasoning, this violation must occur at a node in N_Y because the events $[S(\downarrow Y)]_{N_Y}$ are the latest possible events among events in $S(\downarrow Y)$, in terms of causality.

Therefore, the violation of $\ll(\downarrow Y, X\uparrow)$ can be detected by $|N_X|$ checks for causality between atomic events, by comparing for each i in N_X, $[S(X\uparrow)]_i$ and $[S(\downarrow Y)]_i$. Analogously, the violation of $\ll(\downarrow Y, X\uparrow)$ can be detected by $|N_Y|$ checks for causality between atomic events, by comparing for each i in N_Y, $[S(X\uparrow)]_i$ and $[S(\downarrow Y)]_i$. Therefore, the violation of $\ll(\downarrow Y, X\uparrow)$ can be detected in $min(|N_X|, |N_Y|)$ checks for causality between atomic events.

As noted in Section 2.2, the cuts $C1(X)$, $C2(X)$, $C3(X)$, and $C4(X)$ represent condensed forms of causality information about X, i.e., information about the past and future of the distributed execution associated with X; using the timestamps of these condensed forms of causality information allows the use of timestamps with condensed causality information to derive efficient tests for the relations in Table 1.

Detecting violation of $\ll(\downarrow Y, X\uparrow)$: Applying the timestamps of cuts $\downarrow Y$ and $X\uparrow$ to Key Idea 2, we have the following. The violation of $\ll(\downarrow Y, X\uparrow)$ can be detected by comparing the N_X components of the timestamps of $S(\downarrow Y)$ and $S(X\uparrow)$, or by comparing the N_Y components of the timestamps of $S(\downarrow Y)$ and $S(X\uparrow)$. This leads to the following result. See [8] for proof.

Theorem 19. *$\not\ll(\downarrow Y, X\uparrow)$ can be tested in $min(|N_X|, |N_Y|)$ integer comparisons.*

2.5 Evaluating Causality between Nonatomic Poset Events

As shown informally in Section 2.2, the test for $R(X, Y)$, where $R \in$ Table 1, is equivalent to a test (in some cases, multiple tests) for $\not\ll(\downarrow Y, X\uparrow)$, as indicated in the third column of Table 1. Combining this result with Theorem 19, we have the following result (formally proved in [8]).

Theorem 20. *Each relation $R(X, Y)$ in Table 1 can be evaluated with the following complexity: relations $R1$, $R1'$, $R2'$, $R3$, $R4$, and $R4'$ can be evaluated in $min(|N_X|, |N_Y|)$ integer comparisons, relation $R2$ in $|N_X|$ integer comparisons, and relation $R3'$ in $|N_Y|$ integer comparisons.*

Proof: As shown informally in Section 2.2, the test for $R(X, Y)$, where $R \in$ Table 1, is equivalent to a test (in some cases, multiple tests) for $\not\ll(\downarrow Y, X\uparrow)$, as indicated in the third column of Table 1. The complexity of testing for $\not\ll(\downarrow Y, X\uparrow)$ is as follows.

Relations $R2'$, $R3$, $R4$, $R4'$: These relations can be evaluated using a single test $\downarrow Y \ll X\uparrow$. By Theorem 19, these relations can be evaluated in $min(|N_X|, |N_Y|)$ integer comparisons.

Relation $R2$: This relation can be evaluated using $|N_X|$ tests of the form $\downarrow Y \ll X''\uparrow$, where $|N_{X''}|$= 1. (By using reasoning similar to that at the end of Section 2.3, $|X|$ tests are not needed; testing with only the latest event of X at each node in N_X suffices.) By Theorem 19, each test can be evaluated in 1 integer comparison. So the relation can be evaluated in $|N_X|$ integer comparisons.

Relation $R3'$: This relation can be evaluated using $|N_Y|$ tests of the form $\downarrow Y'' \ll X\uparrow$, where $|N_{Y''}|$= 1. (By using reasoning similar to that at the end of Section 2.3, $|Y|$ tests are not needed; testing with only the earliest event of Y at each node in N_Y suffices.) By Theorem 19, each test can be evaluated in 1 integer comparison. So the relation can be evaluated in $|N_Y|$ integer comparisons.

Relations $R1$, $R1'$: By reasoning similar to that for $R2$ and $R3'$, these relations can be evaluated in $|N_X|$ and also in $|N_Y|$ integer comparisons, i.e., $min(|N_X|, |N_Y|)$ integer comparisons. □

Recall that each of the 32 relations $r(X, Y)$, for $r \in \mathcal{R}$, is equivalent to a relation $R(\hat{X}, \hat{Y})$, where R belongs to Table 1, by using a suitable quantification of X and Y in Table 1 to represent their various proxies $\hat{X}$ and $\hat{Y}$ instead. Each of the 2 proxies of a nonatomic event has 4 cuts associated with it. Thus, Fig. 3 illustrates the four cuts associated with the two proxies of the event X of Fig. 2. The surfaces of the cuts are marked as in Fig. 2. These cuts are used in Theorem 19 upon which Theorem 20 is based. Therefore, Theorem 20 gives the evaluation complexity of each relation $r \in \mathcal{R}$.

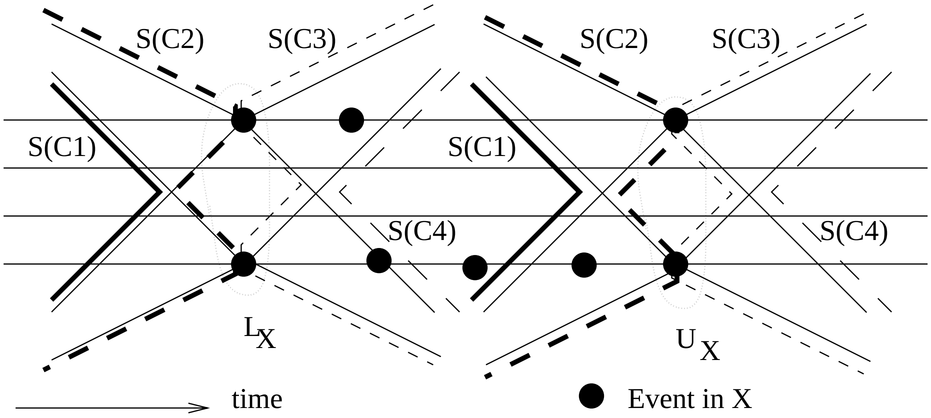

Fig. 3. Cuts of proxies L_X and U_X. X contains 8 atomic events.

It is shown in [8] that the overhead of setting up the timestamp structure is negligible in comparison with the overhead of the evaluation conditions themselves. To address Problem 1, we simply apply the linear-time evaluation conditions in the third column of Table 1 to the various causality relations in the second column, after quantifying X and Y by their appropriate proxies, to evaluate relations in $\mathcal{R}$.

3 Conclusion

The set of causality relations between nonatomic poset events proposed in [11, 12] is useful to distributed real-time applications that need a fine level of discrimination in specifying synchronization conditions. We derived efficient evaluation conditions for these causality relations between nonatomic poset events X and Y; most relations can be evaluated in $min(|N_X|, |N_Y|)$ integer comparisons, some in $|N_X|$ integer comparisons, and the others in $|N_Y|$ integer comparisons, where $|N_X|$ and $|N_Y|$, respectively, are the number of nodes on which the two nonatomic events X and Y occur. The simplified evaluation conditions for the relations have only a linear complexity of testing, whereas a naive evaluation of the relations as per their definitions has a polynomial complexity ($|N_X| \times |N_Y|$) of testing.

During the derivation of our efficient testing conditions, we also defined special system execution prefixes associated with nonatomic poset events. We also saw how to capture causality information associated with a nonatomic event, i.e., information about the past and future execution associated with the nonatomic event, in a condensed and aggregated form via the definition of special execution prefixes associated with the nonatomic event. Furthermore, we provided a mechanism to capture such condensed information about causality of a nonatomic event using a timestamp that has the same size as the timestamp of a single atomic event. As distributed real-time applications become more widespread [11, 12], the proposed theory will be useful to evaluate causality relations between distributed nonatomic events.

References

1. J.W. de Bakker, W.P. de Roever, G. Rozenberg (Eds.), *Linear Time, Branching Time, and Partial Orders in Logics and Models of Concurrency*, LNCS 354, Springer-Verlag, 1989.
2. J. van Benthem, *The Logic of Time*, Kluwer Academic Publishers, (1ed. 1983), 2ed. 1991.
3. K. M. Chandy, J. Misra, *How Processes Learn*, Distributed Computing, 1, 40-52, 1986.
4. C. Fidge, Timestamps in message-passing systems that preserve partial ordering, *Australian Computer Science Communications*, Vol. 10, No. 1, 56-66, February 1988.
5. P.C. Fishburn, *Interval Orders and Interval Graphs: A Study of Partially Ordered Sets,* Wiley & Sons, 1985.
6. F. Jahanian, R. Rajkumar, S. Raju, Run-time monitoring of timing constraints in distributed real-time systems, *Real-Time Systems*, 7(3), 247-273, Nov. 1994.
7. W. Janssen, M. Poel, J. Zwiers, Action systems and action refinement in the development of parallel systems, *Concur'91*, LNCS 527, Springer-Verlag, 298-316, 1991.
8. A. Kshemkalyani, Temporal interactions of intervals in distributed systems, *Tech. Report 29.1933, IBM*, Sept. 1994.
9. A. Kshemkalyani, Temporal interactions of intervals in distributed systems, *Journal of Computer and System Sciences*, 52(2), 287-298, April 1996 (contains some parts of [8]).
10. A. Kshemkalyani, A framework for viewing atomic events in distributed computations, *Theoretical Computer Science* (to appear). (Extended abstract appears in *EuroPar'96*, LNCS 1123, Springer-Verlag, 496-505, Aug. 1996.)
11. A. Kshemkalyani, Synchronization for distributed real-time applications, *5th Workshop on Parallel and Distributed Real-time Systems*, IEEE Computer Society Press, 81-90, April 1997.
12. A. Kshemkalyani, Relative timing constraints between complex events, *Proc. 8th IASTED Conf. on Parallel and Distributed Computing and Systems*, 324-326, Oct. 1996.
13. A. Kshemkalyani, Causality between nonatomic poset events in distributed computations, *IEEE Workshop on Future Trends in Distributed Computing Systems*, 276-282, Oct. 1997.
14. L. Lamport, Time, clocks, and the ordering of events in a distributed system, *Communications of the ACM*, 558-565, 21(7), July 1978.
15. L. Lamport, On interprocess communication, Part I: Basic formalism, Part II: Algorithms, *Distributed Computing*, 1:77-101, 1986.
16. F. Mattern, Virtual time and global states of distributed systems, *Parallel and Distributed Algorithms*, North-Holland, 215-226, 1989.
17. E.R. Olderog, *Nets, Terms, and Formulas,* Cambridge Tracts in Theoretical Computer Science, 1991.
18. A. Rensink, *Models and Methods for Action Refinement,* Ph.D. thesis, University of Twente, The Netherlands, Aug. 1993.
19. R. Rajkumar, *Synchronization in Real-Time Systems,* Kluwer Academic Press, 1991.
20. R. Schwarz, F. Mattern, Detecting causal relationships in distributed computations: In search of the holy grail, *Distributed Computing*, 7:149-174, 1994.

This article was processed using the LaTeX macro package with LLNCS style