Analysis of Interval-Based Global State Detection

Punit Chandra and Ajay D. Kshemkalyani

Computer Science Department, Univ. of Illinois at Chicago, Chicago, IL 60607, USA
{pchandra, ajayk}@cs.uic.edu

Abstract. The problem of global state observation is fundamental to distributed systems. All interactions in distributed systems can be analyzed in terms of the building block formed by the pairwise interactions of intervals between two processes. Considering causality-based pairwise interactions by which two intervals at different processes may interact with each other, there are 40 possible orthogonal interactions. This paper examines the problem: "If a global state of interest to an application is specified in terms of the pairwise interaction types between each pair of processes, how can such a global state be detected?" A solution identifies a global state in which the relation specified for each process pair is satisfied. This paper formulates the specific conditions on the exact communication structures to determine which of the intervals being examined at any time may never satisfy the stipulated relation for that pair of processes, and therefore that interval must be deleted.

1 Introduction

The problem of global state observation is fundamental to distributed systems, as identified by Chandy and Lamport's seminal paper on recording global states [6]. It has been observed that all causality-based interactions in distributed systems can be analyzed in terms of the building block formed by the pairwise interactions of intervals between two processes [11]. A detailed analysis of the causality-based pairwise interactions by which two processes may interact with each other identified 29 (40) causality-based orthogonal interactions, denoted as $\Re$, between two processes under the dense (and nondense) time model, respectively [11]. This paper examines the state detection problem: "If a global state of interest to an application is specified in terms of the pairwise interaction types between each pair of processes, how can such a global state be detected?"

Central to the pairwise interactions studied in this paper is the notion of *time intervals* at each process. A time interval at a process is the local duration in which the process "interacts", or in which some *local* property of interest is true. The semantics of the interval are application-dependent [8, 9, 11, 12, 15, 18]; application areas such as sensor networks, distributed debugging, deadlock characterization [16], predicate detection [3, 4, 5], checkpointing [7, 10], and industrial process control model such intervals.

The above state detection problem was formulated as the following problem DOOR for the Detection of Orthogonal Relations [1, 12].

G. Chakraborty (Ed.): ICDCIT 2005, LNCS 3816, pp. 203–216, 2005.

Problem DOOR. Given a relation $r_{i,j}$ from $\Re$ for each pair of processes i and j, devise a distributed on-line algorithm to identify the intervals, if they exist, one from each process, such that each relation $r_{i,j}$ is satisfied by the (i, j) process pair.

A solution satisfying the set of relations $\{r_{i,j}(\forall i, j)\}$ identifies a global state of the system [6, 14]. We showed [3] that this problem generalizes the global predicate detection problem [4, 5], and further that the solution to this problem is not more expensive than existing solutions to global predicate detection.

Devising an efficient on-line algorithm to solve problem DOOR is a challenging problem because of the overhead of having to track the intervals at different processes. Three solutions have been proposed to this problem so far. A distributed on-line algorithm to solve this problem was outlined in [1]. This algorithm uses $O(n \cdot \min(np, 4mn))$ number of messages with a message size of $O(n)$, where n is the number of processes, m is the maximum number of messages sent by any process, and p is the maximum number of intervals at any process. Another distributed algorithm requiring fewer messages, but at the cost of somewhat larger messages, was given in [2]. This algorithm uses $O(\min(np, 4mn))$ number of messages with a message size of $O(n^2)$. For both the algorithms, the total space complexity across all the processes is $\min(4n^2p - 2np, 10n^2m)$, and the average time complexity at a process is $O(\min(np, 4mn))$. A centralized on-line algorithm run at a server P_0 was given in [3]. For this algorithm, M = maximum queue length at P_0, $p \geq M$ as all the intervals may not be sent to P_0. The performance of the algorithms is summarized in Table 1.

Summary of Results and Contributions. The algorithms in [1, 2, 3] to solve **DOOR** were presented without any formal discussion or analysis of the theoretical basis, and without any correctness proofs. This paper makes the following contributions.

1. To devise any efficient solution, this paper formulates specific conditions on the structure of the exact causal communication patterns to determine which

Table 1. Comparison of space, message and time complexities

Centralized algorithm	Avg. time complexity at P_0	Total number of messages	Space at P_0 (= total msg. space)	Avg. space at P_i, $i \in [1, n]$
Fine_Rel [3]	$O(n^2M)$ or $O(min(n^2p, 4mn^2))$	$O(min(np, 4mn))$	$O(min((4n-2)np, 10n^2m))$	$O(n)$

Distributed Algorithms	Average time complexity/ proc.	Total number of messages	Message size	Total space
Algorithm [1]	$O(min(np, 4mn))$	$O(n \cdot min(np, 4mn))$	$O(n)$	$O(min(2np(2n-1), 10n^2m))$
Algorithm [2]	$O(min(np, 4mn))$	$O(min(np, 4mn))$	$O(n^2)$	$O(min(2np(2n-1), 10n^2m))$

of two intervals being examined from processes i and j may never satisfy $r_{i,j}$, and therefore that interval(s) must be deleted. This result is embodied as:
- a basic principle that we prove in Theorem 1 – the main result, and
- Lemma 4, a useful lemma derived from the above theorem, and used by the algorithms in [1, 2, 3], that can be used to efficiently manage the distributed data structures.

The on-line algorithms [1, 2, 3] to solve problem DOOR indirectly used Lemma 4, but did not explain the principle or indicate how it was derived. This paper derives and explains the critical principle (Theorem 1) from scratch. Any future algorithms to solve DOOR will also have to be based on this principle.
2. Global state observation [6] and predicate detection [4, 5] are fundamental problems. The result provides an understanding of interval-based global state observation and predicate detection, in terms of the causal communication structure in an execution [15].
3. The process of devising this principle (Theorem 1) which guarantees that at least one of any pair of intervals being examined at any time can be deleted (Lemma 4), gives a deeper insight into the nature of reasoning with the structure of causality in a distributed execution. Schwarz and Mattern have identified this as an important problem [19].

Section 2 reviews the background. Section 3 gives the theory used to determine which of two given intervals at different processes can never be part of a solution set, thus allowing at least one of them to be deleted. Section 4 gives concluding remarks.

2 System Model and Background

We assume an asynchronous distributed system in which n processes communicate by reliable message passing over logical FIFO channels [11, 18]. A poset event structure $(E, \prec)$, where $\prec$ is an irreflexive partial ordering representing the causality or the "happens before" relation [17] on the event set E, is used as the model for the execution. E is partitioned into local executions at each process. Each E_i is a linearly ordered set of events executed by process P_i. An event e executed by P_i is denoted e_i. The set of processes is denoted by N.

A *cut* C is a subset of E such that if $e_i \in C$ then $(\forall e'_i)\, e'_i \prec e_i \implies e'_i \in C$. A *consistent cut* is a downward-closed subset of E and denotes an execution prefix. For event e, there are two special consistent cuts $\downarrow e$ and $e \uparrow$, defined next.

Definition 1. *Cut* $\downarrow e$ *is the maximal set of events* $\{e' \,|e' \prec e\}$ *that happen before* e. *Cut* $e \uparrow$ *is the set of all events* $\{e' \,|e' \not\succeq e\} \bigcup \{e_i, i = 1, \ldots, |N| \mid e_i \succeq e \bigwedge (\forall e'_i \prec e_i, e'_i \not\succeq e)\}$ *up to and including the earliest events at each process for which* e *happens before the events.*

The system state after the events in a cut is a *global state* [6]; if the cut is consistent, the corresponding system state is a consistent global state. The durations

Table 2. Dependent relations for interactions between intervals [11]

Relation r	Expression for $r(X, Y)$
R1	$\forall x \in X \forall y \in Y, x \prec y$
R2	$\forall x \in X \exists y \in Y, x \prec y$
R3	$\exists x \in X \forall y \in Y, x \prec y$
R4	$\exists x \in X \exists y \in Y, x \prec y$
S1	$\exists x \in X \forall y \in Y, x \not\preceq y \bigwedge y \not\preceq x$
S2	$\exists x_1, x_2 \in X \exists y \in Y, x_1 \prec y \prec x_2$

Table 3. The 40 orthogonal relations in $\Re$ [11]. The upper part gives the 29 relations assuming dense time. The lower part gives 11 additional relations for nondense time.

Interaction Type	Relation $r(X, Y)$						Relation $r(Y, X)$					
	R1	R2	R3	R4	S1	S2	R1	R2	R3	R4	S1	S2
$IA(= IQ^{-1})$	1	1	1	1	0	0	0	0	0	0	0	0
$IB(= IR^{-1})$	0	1	1	1	0	0	0	0	0	0	0	0
$IC(= IV^{-1})$	0	0	1	1	1	0	0	0	0	0	0	0
$ID(= IX^{-1})$	0	0	1	1	1	1	0	1	0	1	0	0
$ID'(= IU^{-1})$	0	0	1	1	0	1	0	1	0	1	0	1
$IE(= IW^{-1})$	0	0	1	1	1	1	0	0	0	1	0	0
$IE'(= IT^{-1})$	0	0	1	1	0	1	0	0	0	1	0	1
$IF(= IS^{-1})$	0	1	1	1	0	1	0	0	0	1	0	1
$IG(= IG^{-1})$	0	0	0	0	1	0	0	0	0	0	1	0
$IH(= IK^{-1})$	0	0	0	1	1	0	0	0	0	0	1	0
$II(= IJ^{-1})$	0	1	0	1	0	0	0	0	0	0	1	0
$IL(= IO^{-1})$	0	0	0	1	1	1	0	1	0	1	0	0
$IL'(= IP^{-1})$	0	0	0	1	0	1	0	1	0	1	0	1
$IM(= IM^{-1})$	0	0	0	1	1	0	0	0	0	1	1	0
$IN(= IM'^{-1})$	0	0	0	1	1	1	0	0	0	1	0	0
$IN'(= IN'^{-1})$	0	0	0	1	0	1	0	0	0	1	0	1
$ID''(= (IUX)^{-1})$	0	0	1	1	0	1	0	1	0	1	0	0
$IE''(= (ITW)^{-1})$	0	0	1	1	0	1	0	0	0	1	0	0
$IL''(= (IOP)^{-1})$	0	0	0	1	0	1	0	1	0	1	0	0
$IM''(= (IMN)^{-1})$	0	0	0	1	0	0	0	0	0	1	1	0
$IN''(= (IMN')^{-1})$	0	0	0	1	0	1	0	0	0	1	0	0
$IMN''(= (IMN'')^{-1})$	0	0	0	1	0	0	0	0	0	1	0	0

of interest at each process are the durations during which the process interacts, or during which the local application-specific predicate is true. Such a duration, also termed as an *interval*, at process P_i is identified by the corresponding events within E_i. Each interval can be viewed as defining an event of higher granularity at that process, as far as the local predicate of interest is concerned. Such higher-level events, one from each process, can be used to identify a global state [8, 13]. Intervals are denoted by capitals such as X. An interval X at P_i is also denoted by X_i.

It has been shown that there are 29 or 40 possible mutually *orthogonal* ways in which any two durations can be related to each other, depending on whether the *dense* or the *nondense* time model is assumed [11]. Informally speaking, with dense time, $\forall x, y$ in interval A, $x \prec y \Longrightarrow \exists z \in A \mid x \prec z \prec y$. These orthogonal interaction types were identified by first using the six relations given in the

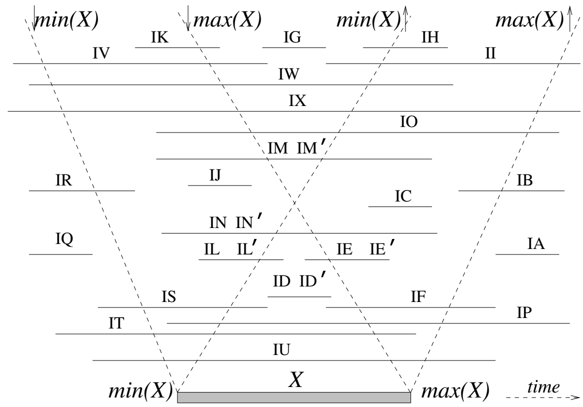


Fig. 1. Interaction types between intervals under the dense time model [11]

first two columns of Table 2. Relations R1 (strong precedence), R2 (partially strong precedence), R3 (partially weak precedence), R4 (weak precedence) define *causality conditions* whereas S1 and S2 define *coupling conditions.*

– (Dense time:) The 29 possible interaction types between a pair of intervals are given in the upper part of Table 3. The interaction types are specified using boolean vectors. The six relations R1-R4 and S1-S2 form a boolean vector of length 12, (six bits for $r(X, Y)$ and six bits for $r(Y, X)$). Of the 29 interactions, there are 13 pairs of inverses, while three are inverses of themselves. The interaction types are illustrated in Figure 1, where interval X is shown by a rectangle. Interval Y, indicated using horizontal lines, is in different positions relative to X. Each position of Y is labeled by an interaction type, IA through IX. The different types of interactions are identified by the various positions of Y relative to X. Five positions of Y have two labels each – the distinction between them is given in [11].
– (Nondense time:) The nondense time model which captures the reality that event sequences and real clocks are discrete permits 11 interaction types between a pair of intervals, defined in the lower part of Table 3, in addition to the 29 identified before. Of these, there are five pairs of inverses, while one is its own inverse. Illustrations are given in [11].

The set of 40 orthogonal relations is denoted as $\Re$.

Example specification of DOOR. Consider a system of three processes P_i, P_j, and P_k. The application wants to detect a global state in which the following relations are pairwise satisfied: (i) $IQ(X_i, Y_j)$ and $IA(Y_j, X_i)$, (ii) $IG(Y_j, Z_k)$ and $IG(Z_k, Y_j)$, and (iii) $IA(Z_k, X_i)$ and $IQ(X_i, Z_k)$.

Each of the 40 orthogonal relations in $\Re$ can be tested for using the bit-patterns for the dependent relations, as given in Table 3. The tests for the relations $R1 - R4$, $S1$, and $S2$ using vector timestamps are given in [1, 2, 3, 12]. During an execution, the information about intervals at P_i is recorded in queue

Q_i. The intervals from the queues are examined pairwise across queues to check if the relation $r_{i,j}$ specified for P_i and P_j holds. In the algorithms in [1, 2], the tests are collectively run in different distributed ways to solve **DOOR**, whereas in the algorithm in [3], they are run at a central server.

To understand the principle for designing these [1, 2, 3] and more efficient algorithms to process the queued intervals, we show our main result (Theorem 1) about when two given intervals may potentially satisfy a given interaction type we want to detect. This theorem in the form of Lemma 4 is used in practice by the algorithms [1, 2, 3] to solve **DOOR**.

3 The Elimination Conditions

Devising an efficient on-line algorithm to solve problem DOOR is a challenge because of the overhead of having to track the intervals at different processes. To devise any efficient solution, we formulate a basic principle that can be used to efficiently manage the distributed data structures. Specifically, we use the notion of a "prohibition" function [1, 2, 3] to show the main principle – Theorem 1 – and thereby Lemma 4 which is the condition for pruning of intervals from queues. We show that if the given relationship between a pair of processes does not hold for a pair of intervals being tested, then at least one of the intervals is deleted.

For any two intervals X and X' that occur at the same process, if $R1(X, X')$, then we say that X is a *predecessor* of X' and X' is a *successor* of X. We assume interval X occurs at P_i and interval Y occurs at P_j. Intuitively, for each $r_{i,j} \in \Re$, a *prohibition function* $\mathcal{H}(r_{i,j})$ is the set of all relations R such that if $R(X, Y)$ is true, then $r_{i,j}(X, Y')$ can never be true for some successor Y' of Y. $\mathcal{H}(r_{i,j})$ is the set of relations that prohibit $r_{i,j}$ from being true in the future.

Definition 2. Prohibition function $\mathcal{H} : \Re \to 2^{\Re}$ *is defined as* $\mathcal{H}(r_{i,j}) = \{R \in \Re \mid if\ R(X, Y)\ is\ true\ then\ r_{i,j}(X, Y')\ is\ false\ for\ all\ Y'\ that\ succeed\ Y\ \}$.

Two relations R' and R'' in $\Re$ are related by the *allows* relation $\rightsquigarrow$ if the occurrence of $R'(X, Y)$ does not prohibit $R''(X, Y')$ for some successor Y' of Y.

Definition 3. *The "allows" relation* $\rightsquigarrow$ *is a relation on* $\Re \times \Re$ *such that* $R' \rightsquigarrow R''$ *if the following holds: if* $R'(X, Y)$ *is true then* $R''(X, Y')$ *can be true for some* Y' *that succeeds* Y.

Lemma 1. *If* $R \in \mathcal{H}(r_{i,j})$ *then* $R \not\rightsquigarrow r_{i,j}$ *else if* $R \notin \mathcal{H}(r_{i,j})$ *then* $R \rightsquigarrow r_{i,j}$.

Proof. If $R \in \mathcal{H}(r_{i,j})$, using Definition 2, it can be inferred that $r_{i,j}$ is false for all Y' that succeed Y. This does not satisfy Definition 3. Hence $R \not\rightsquigarrow r_{i,j}$. If $R \notin \mathcal{H}(r_{i,j})$, it follows that $r_{i,j}$ can be true for some Y' that succeeds Y. This satisfies Definition 3 and hence $R \rightsquigarrow r_{i,j}$. □

Given that $R'(A, B) \rightsquigarrow R''(A, B')$, where R' and R'' are orthogonal relations from $\Re$, the following lemma shows some relationships between interval pairs

A, B and A, B' in terms of the dependent set of causality relations $R1 - R4$. These relationships will be useful to show a critical relationship between R'^{-1} and R''^{-1} (Theorem 1) that allows efficient pruning of intervals on the queues in any algorithm to solve Problem **DOOR**.

Lemma 2. *If $R' \leadsto R''$, $R'(A, B)$ and $R''(A, B')$, where R', $R'' \in \Re$, then the statements in Table 5 are true.*

Proof. As $R' \leadsto R''$ and $R'(A, B)$ is true, we can safely assume that there can exist an interval B' that succeeds B and such that $R''(A, B')$ is true. Now consider axioms AL2, AL4, AL5 and AL6 given in Table 4. Applying the following transformations gives statements T1 to T4 of Table 5, respectively.

1. Substitute A, B, B' for X, Y, Z, respectively, in Table 4.
2. As B' succeeds B, hence substitute true for $R1(B, B')$, $R2(B, B')$, $R3(B, B')$, and $R4(B, B')$.

Consider axioms AL1, AL2, AL3 and AL4 given in Table 4. Applying the following transformations gives statements T5 to T8, of Table 5, respectively.

1. Substitute B, B', and A for X, Y, and Z, respectively in Table 4.
2. As B' succeeds B, hence substitute true for $R1(B, B')$, $R2(B, B')$, $R3(B, B')$, and $R4(B, B')$. □

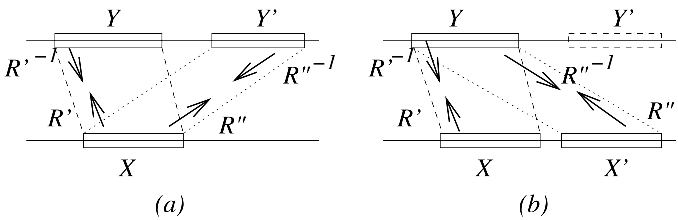
We now show an important result between any two relations in $\Re$ that satisfy the "allows" relation, and the existence of the "allows" relation between their

Table 4. Axioms for the causality relations of Table 2 [11]. $\overline{R}$ stands for "R is $false$".

Axiom Label	$r_1(X,Y) \bigwedge r_2(Y,Z) \Longrightarrow r(X,Z)$
AL1	$R1(X,Y) \bigwedge R2(Y,Z) \Longrightarrow R2(X,Z)$
AL2	$R1(X,Y) \bigwedge R3(Y,Z) \Longrightarrow R1(X,Z)$
AL3	$R1(X,Y) \bigwedge R4(Y,Z) \Longrightarrow R2(X,Z)$
AL4	$R2(X,Y) \bigwedge R1(Y,Z) \Longrightarrow R1(X,Z)$
AL5	$R3(X,Y) \bigwedge R1(Y,Z) \Longrightarrow R3(X,Z)$
AL6	$R4(X,Y) \bigwedge R1(Y,Z) \Longrightarrow R3(X,Z)$
AL7	$R2(X,Y) \bigwedge R3(Y,Z) \Longrightarrow true$
AL8	$R2(X,Y) \bigwedge R4(Y,Z) \Longrightarrow true$
AL9	$R3(X,Y) \bigwedge R2(Y,Z) \Longrightarrow R4(X,Z)$
AL10	$R4(X,Y) \bigwedge R2(Y,Z) \Longrightarrow R4(X,Z)$
AL11	$R3(X,Y) \bigwedge R4(Y,Z) \Longrightarrow R4(X,Z)$
AL12	$R4(X,Y) \bigwedge R3(Y,Z) \Longrightarrow true$
AL13	$R1(X,Y) \Longrightarrow \overline{S1}(X,Y) \bigwedge \overline{S2}(X,Y) \bigwedge \overline{R4}(Y,X) \bigwedge \overline{S1}(Y,X) \bigwedge \overline{S2}(Y,X)$
AL14	$R2(X,Y) \Longrightarrow \overline{S1}(X,Y) \bigwedge \overline{R2}(Y,X)$
AL15	$R3(X,Y) \Longrightarrow \overline{R3}(Y,X) \bigwedge \overline{S1}(Y,X)$
AL16	$R4(X,Y) \Longrightarrow \overline{R1}(Y,X)$
AL17	$S1(X,Y) \Longrightarrow \overline{R2}(X,Y) \bigwedge \overline{R3}(Y,X) \bigwedge \overline{S2}(Y,X)$
AL18	$S2(X,Y) \Longrightarrow \overline{R1}(X,Y) \bigwedge R4(X,Y) \bigwedge \overline{R1}(Y,X) \bigwedge R4(Y,X) \bigwedge \overline{S1}(Y,X)$

Table 5. Given $R' \leadsto R''$, $R'(A, B)$ and $R''(A, B')$, for R', $R'' \in \Re$, statements between interval pairs A, B and A, B' using the dependent relations $R1 - R4$

Statement Label	Statements
T1	$R1(A, B) \Longrightarrow R1(A, B')$
T2	$R2(A, B) \Longrightarrow R1(A, B')$
T3	$R3(A, B) \Longrightarrow R3(A, B')$
T4	$R4(A, B) \Longrightarrow R3(A, B')$
T5	$R1(B', A) \Longrightarrow R1(B, A)$
T6	$R2(B', A) \Longrightarrow R2(B, A)$
T7	$R3(B', A) \Longrightarrow R1(B, A)$
T8	$R4(B', A) \Longrightarrow R2(B, A)$



(a) R'(X,Y), R"(X,Y'), and hence, R' allows R"

(b) From (a) we have $R'^{-1}(Y,X)$, $R''^{-1}(Y',X)$. *But can* $R''^{-1}(Y,X')$ *hold? Theorem shows it cannot. Hence.* R'^{-1} *does not allow* R''^{-1}

Fig. 2. Illustration of Theorem 1

respective inverses. Specifically, if R' allows R'' (and $R' \neq R''$), then Theorem 1 shows that R'^{-1} necessarily does not allow relation R''^{-1}. This theorem is illustrated in Figure 2. This theorem is used in deriving Lemma 4 which will be practically used in deriving solutions to problem DOOR, and to prove the correctness of such solutions.

Theorem 1. *For* $R', R'' \in \Re$ *and* $R' \neq R''$, *if* $R' \leadsto R''$ *then* $R'^{-1} \not\leadsto R''^{-1}$

Proof. We prove by contradiction. The assumption using which we show a contradiction is the following.

$$R'(X, Y) \texttt{ is true, } R'(X, Y) \leadsto R''(X, Y') \texttt{ and } R'^{-1}(Y, X) \leadsto R''^{-1}(Y, X') \tag{1}$$

As T1 to T8 must hold for both $R'(X, Y) \leadsto R''(X, Y')$ and $R'^{-1}(Y, X) \leadsto R''^{-1}(Y, X')$ we get two sets of constraints for intervals X, X', Y, and Y' in terms of the dependent causality relations $R1$ to $R4$.

Consider $R'(X, Y) \leadsto R''(X, Y')$. Instantiating A by X, B by Y, and B' by Y' in T1-T8, we have the following set of constraints that need to be satisfied.

C1: $R1(X,Y) \Rightarrow R1(X,Y')$
C2: $R2(X,Y) \Rightarrow R1(X,Y')$
C3: $R3(X,Y) \Rightarrow R3(X,Y')$
C4: $R4(X,Y) \Rightarrow R3(X,Y')$
C5: $R1(Y',X) \Rightarrow R1(Y,X)$
C6: $R2(Y',X) \Rightarrow R2(Y,X)$
C7: $R3(Y',X) \Rightarrow R1(Y,X)$
C8: $R4(Y',X) \Rightarrow R2(Y,X)$

Now consider $R'^{-1}(Y,X) \rightsquigarrow R''^{-1}(Y,X')$. Instantiating A by Y, B by X, and B' by X' in T1-T8, we have the following set of constraints that need to be satisfied.

C9: $R1(Y,X) \Rightarrow R1(Y,X')$
C10: $R2(Y,X) \Rightarrow R1(Y,X')$
C11: $R3(Y,X) \Rightarrow R3(Y,X')$
C12: $R4(Y,X) \Rightarrow R3(Y,X')$
C13: $R1(X',Y) \Rightarrow R1(X,Y)$
C14: $R2(X',Y) \Rightarrow R2(X,Y)$
C15: $R3(X',Y) \Rightarrow R1(X,Y)$
C16: $R4(X',Y) \Rightarrow R2(X,Y)$

From Equation 1, it can be seen that the interval pairs (Y',X) and (Y,X') both are related by the orthogonal relation R''^{-1}. Hence $r(Y',X) \Leftrightarrow r(Y,X')$, where r is any of the six dependent relations given in Table 2. Thus replacing $r(Y,X')$ by $r(Y',X)$ in C9 to C12, we have the following constraints.

C17: $R1(Y,X) \Rightarrow R1(Y',X)$
C18: $R2(Y,X) \Rightarrow R1(Y',X)$
C19: $R3(Y,X) \Rightarrow R3(Y',X)$
C20: $R4(Y,X) \Rightarrow R3(Y',X)$

From Equation 1, it can also be seen in a similar way that the interval pairs (X,Y') and (X',Y) both are related by the orthogonal relation R''. Hence $r(X,Y') \Leftrightarrow r(X',Y)$, where r is any of the six dependent relations given in Table 2. Thus replacing $r(X',Y)$ by $r(X,Y')$ in C13 to C16, we have the following constraints.

C21: $R1(X,Y') \Rightarrow R1(X,Y)$
C22: $R2(X,Y') \Rightarrow R2(X,Y)$
C23: $R3(X,Y') \Rightarrow R1(X,Y)$
C24: $R4(X,Y') \Rightarrow R2(X,Y)$

The two constraint sets (C1)-(C8) and (C17)-(C24) given above can be combined to obtain restrictions on the type of interactions (given in Table 3) that $R'(X,Y)$ can belong to. Combining constraints C1 to C4 with constraints C21 to C24 gives

$$R1(X,Y) \vee R2(X,Y) \vee R3(X,Y) \vee R4(X,Y) \Rightarrow R1(X,Y)$$

Note from the definitions in Table 2 that $R1(X,Y) \Rightarrow R2(X,Y) \wedge R3(X,Y) \wedge R4(X,Y)$. Thus,

$$\begin{gathered} R1(X,Y) \vee R2(X,Y) \vee R3(X,Y) \vee R4(X,Y) \Rightarrow \\ R1(X,Y) \wedge R2(X,Y) \wedge R3(X,Y) \wedge R4(X,Y) \end{gathered} \tag{2}$$

The above implication implies that either relations $R1(X,Y)$, $R2(X,Y)$, $R3(X,Y)$, and $R4(X,Y)$ are all true or all false.

Using a similar approach, combining constraints C17 to C20 with constraints C5 to C8 gives

$$\begin{gathered} R1(Y,X) \vee R2(Y,X) \vee R3(Y,X) \vee R4(Y,X) \Rightarrow \\ R1(Y,X) \wedge R2(Y,X) \wedge R3(Y,X) \wedge R4(Y,X) \end{gathered} \tag{3}$$

This means either relations $R1(Y,X)$, $R2(Y,X)$, $R3(Y,X)$, and $R4(Y,X)$, are all true or all false.

Implications (2) and (3) restrict the interaction type (given in Table 3) to which $R'(X,Y)$ can belong. We now examine all the restricted cases to which $R'(X,Y)$ can belong, i.e., when $R1(X,Y)$ to $R4(X,Y)$ are all true, and when $R1(X,Y)$ to $R4(X,Y)$ are all false, and show that $R'(X,Y)$ can not exist; which is a contradiction to Equation (1).

Case 1. $R1(X,Y)$, $R2(X,Y)$, $R3(X,Y)$, and $R4(X,Y)$ are all true.

From constraints C1 to C4, we get

$$R1(X,Y'),\ R2(X,Y'),\ R3(X,Y'),\ R4(X,Y')\ \texttt{are true.} \tag{4}$$

Using axioms AL13 to AL16 we get $R1(Y,X)$, $R2(Y,X)$, $R3(Y,X)$, $R4(Y,X)$, $S1(X,Y)$, $S2(X,Y)$, $S1(Y,X)$, $S2(Y,X)$ are all false. Now substituting X, Y' for X, Y in axioms AL13 to AL16, we get

$$\begin{gathered} R1(Y',X),\ R2(Y',X),\ R3(Y',X),\ R4(Y',X),\ S1(X,Y'),\ S2(X,Y'),\\ S1(Y',X),\ S2(Y',X)\ \texttt{are false.} \end{gathered} \tag{5}$$

Using Table 3, the only possible combination by which to instantiate R' and R'' so that they satisfy Equations (4) and (5) is IA. Thus, we have $R'(X,Y) = R''(X,Y') = IA$. As $R' \neq R''$ by the theorem statement, this case cannot exist.

Case 2. $R1(X,Y)$, $R2(X,Y)$, $R3(X,Y)$ and $R4(X,Y)$ are all false.

This case has two subcases.

1. $R1(Y,X)$, $R2(Y,X)$, $R3(Y,X)$, and $R4(Y,X)$ are all true. From constraints C17 to C20, we get

$$R1(Y',X),\ R2(Y',X),\ R3(Y',X),\ R4(Y',X)\ \texttt{are true.} \tag{6}$$

   Substituting Y, X for X, Y in axiom AL13 we get $S1(X,Y)$, $S2(X,Y)$, $S1(Y,X)$, $S2(Y,X)$, are all false. Now substituting Y', X for X, Y in axioms AL13 to AL16, we get

$$\begin{gathered} R1(X,Y'),\ R2(X,Y'),\ R3(X,Y'),\ R4(X,Y'),\ S1(X,Y'),\ S2(X,Y'),\\ S1(Y',X),\ S2(Y',X)\ \texttt{are false.} \end{gathered} \tag{7}$$

   Using Table 3, the only possible combination by which to instantiate R' and R'' so that they satisfy Equations (6) and (7) is IQ. Thus, we have $R'(X,Y) = R''(X,Y') = IQ$. As $R' \neq R''$ by the theorem statement, this case cannot exist.
2. $R1(Y,X)$, $R2(Y,X)$, $R3(Y,X)$, and $R4(Y,X)$ are all false. From constraints C5 to C8, we get

$$R1(Y',X),\ R2(Y',X),\ R3(Y',X),\ R4(Y',X)\ \texttt{are false.} \tag{8}$$

Table 6. $\mathcal{H}(r_{i,j})$ for the 40 independent relations in $\Re$. The upper part gives function $\mathcal{H}$ for dense time. The lower part gives the function $\mathcal{H}$ for the 11 additional relations for non-dense time.

Interaction Type $r_{i,j}$	$\mathcal{H}(r_{i,j})$	$\mathcal{H}(r_{j,i})$
$IA\ (= IQ^{-1})$	ϕ	$\Re - \{IQ\}$
$IB\ (= IR^{-1})$	$\{IA, IB, IF, II, IP, IO, IU, IX, IUX, IOP\}$	$\Re - \{IQ\}$
$IC\ (= IV^{-1})$	$\{IA, IB, IF, II, IP, IO, IU, IX, IUX, IOP\}$	$\Re - \{IQ\}$
$ID\ (= IX^{-1})$	$\Re - \{IQ, IS, IR, IJ, IL, IL', IL'', ID, ID', ID''\}$	$\Re - \{IQ\}$
$ID'\ (= IU^{-1})$	$\Re - \{IQ, IS, IR, IJ, IL, IL', IL'', ID, ID', ID''\}$	$\Re - \{IQ\}$
$IE\ (= IW^{-1})$	$\Re - \{IQ, IS, IR, IJ, IL, IL', IL'', ID, ID', ID''\}$	$\Re - \{IQ\}$
$IE'\ (= IT^{-1})$	$\Re - \{IQ, IS, IR, IJ, IL, IL', IL'', ID, ID', ID''\}$	$\Re - \{IQ\}$
$IF\ (= IS^{-1})$	$\Re - \{IQ, IS, IR, IJ, IL, IL', IL'', ID, ID', ID''\}$	$\Re - \{IQ\}$
$IG\ (= IG^{-1})$	$\Re - \{IQ, IR, IJ, IV, IK, IG\}$	$\Re - \{IQ, IR, IJ, IV, IK, IG\}$
$IH\ (= IK^{-1})$	$\Re - \{IQ, IR, IJ, IV, IK, IG\}$	$\Re - \{IQ, IR, IJ\}$
$II\ (= IJ^{-1})$	$\Re - \{IQ, IR, IJ, IV, IK, IG\}$	$\Re - \{IQ, IR, IJ\}$
$IL\ (= IO^{-1})$	$\Re - \{IQ, IR, IJ\}$	$\Re - \{IQ, IR, IJ\}$
$IL'\ (= IP^{-1})$	$\Re - \{IQ, IR, IJ\}$	$\Re - \{IQ, IR, IJ\}$
$IM\ (= IM^{-1})$	$\Re - \{IQ, IR, IJ\}$	$\Re - \{IQ, IR, IJ\}$
$IN\ (= IM'^{-1})$	$\Re - \{IQ, IR, IJ\}$	$\Re - \{IQ, IR, IJ\}$
$IN'\ (= IN'^{-1})$	$\Re - \{IQ, IR, IJ\}$	$\Re - \{IQ, IR, IJ\}$
$ID''\ (= (IUX)^{-1})$	$\Re - \{IQ, IS, IR, IJ, IL, IL', IL'', ID, ID', ID''\}$	$\Re - \{IQ\}$
$IE''\ (= (ITW)^{-1})$	$\Re - \{IQ, IS, IR, IJ, IL, IL', IL'', ID, ID', ID''\}$	$\Re - \{IQ\}$
$IL''\ (= (IOP)^{-1})$	$\Re - \{IQ, IR, IJ\}$	$\Re - \{IQ, IR, IJ\}$
$IM''\ (= (IMN)^{-1})$	$\Re - \{IQ, IR, IJ\}$	$\Re - \{IQ, IR, IJ\}$
$IN''\ (= (IMN')^{-1})$	$\Re - \{IQ, IR, IJ\}$	$\Re - \{IQ, IR, IJ\}$
$IMN''\ (= (IMN'')^{-1})$	$\Re - \{IQ, IR, IJ\}$	$\Re - \{IQ, IR, IJ\}$

Now substituting Y', X for X, Y in axioms AL13 to AL16, we get

$$R1(X,Y'),\ R2(X,Y'),\ R3(X,Y'),\ R4(X,Y')\ \texttt{are false}. \tag{9}$$

Using Table 3, the only possible combination by which to instantiate R' and R'' so that they satisfy Equations (8)-(9) is IG. Thus, we have $R'(X,Y) = R''(X,Y') = IG$. As $R' \neq R''$ by the theorem statement, this case cannot exist.

Hence there cannot exist a case where $R'(X,Y) \rightsquigarrow R''(X,Y')$ and $R'^{-1}(Y,X) \rightsquigarrow R''^{-1}(Y',X)$. This contradicts the assumption in Equation 1, proving the theorem. $\square$

Example. $IC \rightsquigarrow IB \Rightarrow IV(= IC^{-1}) \not\rightsquigarrow IR(= IB^{-1})$, which is indeed *true*. Note that $R' \neq R''$ in the statement of Theorem 1 is necessary; otherwise $R' \rightsquigarrow R'$ leads to $R'^{-1} \not\rightsquigarrow R'^{-1}$ from the theorem, a contradiction.

Table 6 gives $S(r_{i,j})$ for each of the 40 interaction types in $\Re$. The table is constructed by analyzing each interaction pair in $\Re$. The following two lemmas are necessary to show the correctness of the algorithm in [1, 2, 3] and of any other algorithm to solve problem DOOR.

Lemma 3. *If the relationship $R(X,Y)$ between intervals X and Y (belonging to process P_i and P_j, resp.) is contained in the set $\mathcal{H}(r_{i,j})$, and $r_{i,j} \neq R$, then interval X can be removed from the queue Q_i.*

Proof. From the definition of $\mathcal{H}(r_{i,j})$, we get that $r_{i,j}(X, Y')$ cannot exist, where Y' is any successor interval of Y. Further, as $r_{i,j} \neq R$, we have that interval X can never be a part of the solution and can be deleted from the queue. □

The following final result, although simple in form, is based on the crucial Theorem 1 and shows that both $R \notin \mathcal{H}(r_{i,j})$ and $R^{-1} \notin \mathcal{H}(r_{j,i})$ cannot hold when $R \neq r_{i,j}$. Hence, by Lemma 3, if $R(X_i, Y_j) \neq r_{i,j}$ then at least one of the intervals X_i and Y_j being tested must be deleted.

Lemma 4. *If the relationship between a pair of intervals X and Y (belonging to processes P_i and P_j respectively) is not equal to $r_{i,j}$, then interval X or interval Y is removed from its queue Q_i or Q_j, respectively.*

Proof. We use contradiction. Assume relation $R(X, Y)$ ($\neq r_{i,j}(X, Y)$) is true for intervals X and Y. From Lemma 3, the only time neither X nor Y will be deleted is when $R \notin \mathcal{H}(r_{i,j})$ and $R^{-1} \notin \mathcal{H}(r_{j,i})$. From Lemma 1, it can be inferred that $R \rightsquigarrow r_{i,j}$ and $R^{-1} \rightsquigarrow r_{j,i}$. As $r_{i,j}^{-1} = r_{j,i}$, we get $R \rightsquigarrow r_{i,j}$ and $R^{-1} \rightsquigarrow r_{i,j}^{-1}$. This is a contradiction as by Theorem 1, $R \rightsquigarrow r_{i,j} \Rightarrow R^{-1} \not\rightsquigarrow r_{i,j}^{-1}$. Hence $R \in \mathcal{H}(r_{i,j})$ or $R^{-1} \in \mathcal{H}(r_{j,i})$, and thus at least one of the intervals will be deleted. □

Observe with reference to Table 6 that it is possible that both intervals being compared need to be deleted, e.g., when $r_{i,j} = IC$ and $R(X, Y) = IU$.

Significance of Theorem 1 and Lemma 4. Lemma 4 embodies a principle that underlies all solutions to problem DOOR. The algorithms given in [1, 2] use this result of Lemma 4 to efficiently manage and prune the local interval queues to solve problem DOOR in a distributed manner. Essentially, they examine the intervals in the queues, a pair of intervals from different processes, at a time. Lemma 4 guarantees that in each such test, at least one or both intervals being examined are deleted, unless $r_{i,j}(X_i, Y_j)$ is satisfied by that pair of intervals X_i and Y_j. The algorithms differ in the manner in which they construct the queues, and in how they process the intervals and the queues. The algorithm in [3] also relies on this result of Lemma 4 to process the interval information at a central server P_0 in an on-line manner. More efficient solutions to problem DOOR that may arise in the future will also have to use these results.

4 Conclusions

Causality-based pairwise temporal interactions between intervals in a distributed execution provide a valuable way to specify and model synchronization conditions and information interchange. This paper examined the underlying theory to solve the problem (problem DOOR) of how to devise algorithms to identify a set of intervals, one from each process, such that a given set of pairwise temporal interactions, one for each process pair, holds for the set of intervals identified. Devising an efficient on-line algorithm to solve problem DOOR is a challenge because of the overhead of having to track the intervals at different processes. For any two intervals being examined from processes P_i and P_j, this paper formulated and proved the underlying principle which identifies which (or both) of the intervals can be safely deleted if the intervals do not satisfy $r_{i,j}$. This principle can be used by any algorithm, such as those in [1, 2, 3] or any newer algorithms, to efficiently manage the local interval queues to solve problem DOOR.

Problem DOOR is important because it generalizes the global state observation and the predicate detection problems; further, solutions to problem DOOR which provide a much richer palette of information about the causality structure in the application execution (see [3]), cost about the same as the solutions to traditional forms of global predicate detection. The process of formulating the underlying principle of determining which intervals can be discarded as never forming a part of a solution that satisfies a specification of DOOR, also gave a deeper insight into the structure of causality in a distributed execution, and the global state observation and predicate detection problems.

References

1. P. Chandra, A.D. Kshemkalyani, Detection of orthogonal interval relations. Proc. 9th International High Performance Computing Conference (HiPC), LNCS 2552, Springer-Verlag, 323-333, December 2002.
2. P. Chandra, A.D. Kshemkalyani, Global state detection based on peer-to-peer interactions. Proc. IFIP International Conference on Embedded and Ubiquitous Computing (EUC), LNCS, Springer, Dec. 2005.
3. P. Chandra, A.D. Kshemkalyani, Causality-based predicate detection across space and time. IEEE Transactions on Computers, 54(11): 1438-1453, November 2005.
4. B. Charron-Bost, C. Delporte-Gallet, H. Fauconnier, Local and temporal predicates in distributed systems. ACM TOPLAS, 17(1): 157-179, 1995.
5. R. Cooper, K. Marzullo, Consistent detection of global predicates. Proc. ACM/ONR Workshop on Parallel and Distributed Debugging, 163-173, May 1991.
6. K.M. Chandy, L. Lamport, Distributed snapshots: Determining global states of distributed systems. ACM Transactions on Computer Systems, 3(1): 63-75, 1985.
7. E.N. Elnozahy, L. Alvisi, Y.-M. Wang, D.B. Johnson, A survey of rollback-recovery protocols in message-passing systems. ACM Comput. Surv., 34(3): 375-408, 2002.
8. J.M. Helary, A. Mostefaoui, M. Raynal, Virtual precedence in asynchronous systems: Concept and applications. Proc. WDAG, 170-184, LNCS 1320, Springer, 1997.
9. J.M. Helary, A. Mostefaoui, M. Raynal, Interval consistency of asynchronous distributed computations. J. Computer and System Sciences, 64, 329-349, 2002.
10. R. Koo, S. Toueg, Checkpointing and rollback-recovery for distributed systems. IEEE Trans. Software Eng. 13(1): 23-31, 1987.
11. A.D. Kshemkalyani, Temporal interactions of intervals in distributed systems. Journal of Computer and System Sciences, 52(2): 287-298, April 1996.
12. A.D. Kshemkalyani, A fine-grained modality classification for global predicates. IEEE Transactions on Parallel and Distributed Systems, 14(8): 807-816, Aug. 2003.
13. A.D. Kshemkalyani, A framework for viewing atomic events in distributed computations. Theoretical Computer Science, 196(1-2), 45-70, April 1998.
14. A.D. Kshemkalyani, M. Raynal, M. Singhal, An introduction to snapshot algorithms in distributed computing. Distributed Systems Engineering Journal, 2(4): 224-233, 1995.

15. A.D. Kshemkalyani, M. Singhal, Communication patterns in distributed computations. Journal of Parallel and Distributed Computing, 62(6): 1104-1119, June 2002.
16. A. D. Kshemkalyani, M. Singhal, On characterization and correctness of distributed deadlock detection. Journal of Parallel and Distributed Computing, 22(1): 44-59, July 1994.
17. L. Lamport, Time, clocks, and the ordering of events in a distributed system. Communications of the ACM, 558-565, 21(7), July 1978.
18. L. Lamport, On interprocess communication, Part I: Basic formalism; Part II: Algorithms. Distributed Computing, 1:77-85 and 1:86-101, 1986.
19. R. Schwarz, F. Mattern, Detecting causal relationships in distributed computations: In search of the holy grail. Distributed Computing, 7:149-174, 1994.