SIAM J. COMPUT.
Vol. 34, No. 2, pp. 498–513

TEMPORAL REASONING ABOUT TWO CONCURRENT SEQUENCES OF EVENTS*

YASUNORI ISHIHARA†, SHIN ISHII‡, HIROYUKI SEKI‡, AND MINORU ITO‡

Abstract. This paper discusses temporal reasoning with respect to constraints on two concurrent sequences of events. If two given sequences of events can be mapped into one sequence that satisfies a given constraint, then the constraint is said to be consistent. First, we mention that the consistency of such constraints is NP-complete. Then we introduce the notion of graph representations of constraints. If a graph representation of a given constraint c can be constructed in polynomial time, then the consistency of c is decidable in polynomial time. However, it is shown that the graph representability of a given c is coNP-complete. Next, we propose a subclass $\mathrm{CDC}^{\neq}$ of constraints such that for each constraint c in $\mathrm{CDC}^{\neq}$, a graph representation of c can be constructed in polynomial time. The expressive power of $\mathrm{CDC}^{\neq}$ is incomparable to any other subclasses of constraints for which the consistency problem is known to be tractable.

Key words. temporal reasoning, temporal constraint, consistency, complexity

AMS subject classifications. 68Q25, 68Q20

DOI. 10.1137/S0097539799362172

1. Introduction.

In many practical information systems, each fragment of data has temporal information. For example, in relational databases, relations are often augmented by temporal attributes (such databases are called temporal databases [1]). It is often desirable for information systems to manage such temporal information in an intelligent way. For example, suppose that a knowledge base system has the following information:

- Last night, the people in the restaurant heard two shots.
- The electricity was off at least between the first and second shots.
- After the electric power resumed, the people found all the money in the restaurant had been stolen.

We want the knowledge base system to infer that the people found all the money stolen *after* the second shot. This is a trivial but typical example of *temporal reasoning*.

In this paper, temporal reasoning about two concurrent sequences of events is considered. Two sets of time variables $S = \{s_0, s_1, \ldots, s_m\}$ and $T = \{t_0, t_1, \ldots, t_n\}$ are used for describing temporal constraints, where $s_0, s_1, \ldots, s_m$ represent time points of one local clock and $t_0, t_1, \ldots, t_n$ represent time points of the other clock. A temporal constraint consists of expressions of the forms $s_i < t_j$, $s_i > t_j$, $s_i \le t_j$, $s_i \ge t_j$, $s_i = t_j$, and $s_i \neq t_j$, and Boolean operators $\neg$, $\vee$, and $\wedge$. One of the applications of temporal reasoning about such constraints is belief revision [4] in a multiagent environment, as shown in the next example. As far as we know, no paper has focused on the class of constraints such that the number of local clocks is fixed.

Example 1.1. Consider the following multiagent environment: Each agent has its own local clock and records its observations, each of which is a pair consisting

*Received by the editors September 27, 1999; accepted for publication (in revised form) August 18, 2004, published electronically February 3, 2005.
http://www.siam.org/journals/sicomp/34-2/36217.html
†Graduate School of Information Science and Technology, Osaka University, Suita, Osaka, 565-0871 Japan (ishihara@ist.osaka-u.ac.jp).
‡Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Nara, 630-0192 Japan (ishii@is.naist.jp, seki@is.naist.jp, ito@is.naist.jp).

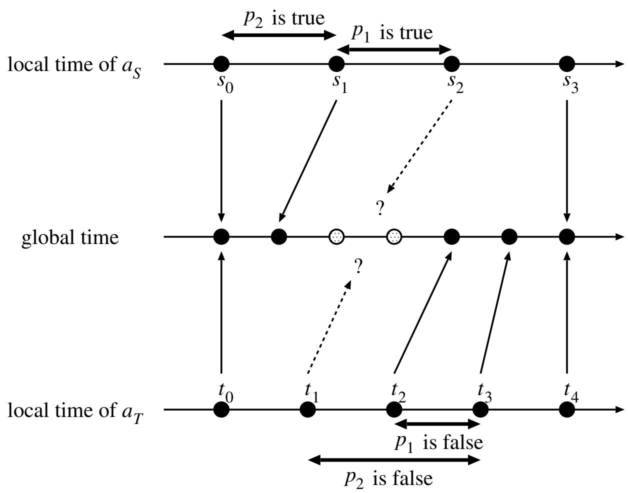


FIG. 1. *Example of temporal reasoning about two concurrent sequences of events.*

of a proposition and an observation time. The agents sometimes meet together and exchange their observations in order to revise and refine their information on the observation times.

See Figure 1. $s_0, \ldots, s_3$ represent local time points of agent a_S with $s_0 < \cdots < s_3$, and $t_0, \ldots, t_4$ represent local time points of agent a_T with $t_0 < \cdots < t_4$. The agents met together at $s_0 = t_0$ and $s_3 = t_4$. Suppose that a_S observed that a proposition p_1 was true during its local time interval $[s_1, s_2]$. Also suppose that a_T observed that p_1 was false during $[t_2, t_3]$. Then the agents can conclude that intervals $[s_1, s_2]$ and $[t_2, t_3]$ are disjoint, i.e., $(s_1 \geq t_3) \vee (s_2 \leq t_2)$. In this paper, a constraint of this form is called a *disjointness constraint* and is denoted by $[s_1, s_2] \not\pitchfork [t_2, t_3]$. The agents also obtain $[s_0, s_1] \not\pitchfork [t_1, t_3]$ from the observations of p_2.

Now, several facts can be inferred from the obtained constraint $c = ([s_1, s_2] \not\pitchfork [t_2, t_3]) \wedge ([s_0, s_1] \not\pitchfork [t_1, t_3])$. For example, $s_1 \leq t_1$ can be concluded since $c \wedge (s_1 > t_1)$ is inconsistent (i.e., unsatisfiable). On the other hand, the order between s_2 and t_1 cannot be determined since all of $c \wedge (s_2 < t_1)$, $c \wedge (s_2 = t_1)$, and $c \wedge (s_2 > t_1)$ are consistent.

Another possible application is job scheduling for two processors, where each job is denoted by a time interval and each pair of mutually exclusive jobs is specified by a disjointness constraint.

The contribution of this paper is as follows. We first mention that the consistency of constraints on two local clocks is NP-complete when both conjunction and disjunction are freely used. Next, we introduce the notion of *graph representations* of constraints. A graph representation of a constraint c is a directed graph representing all the valuations that satisfy c. If c has a graph representation G_c, and G_c can be constructed in polynomial time, then the consistency of c is also decidable in polynomial time. However, it is shown that the graph representability of a given c is coNP-complete (and therefore constructing G_c is coNP-hard). Next, we propose a new tractable subclass CDC$^{\neq}$ of constraints. CDC$^{\neq}$ stands for conjunctive disjointness constraints with inequalities, and it can express conjunctions of constraints in the form of $[s_i, s_{i'}] \not\pitchfork [t_j, t_{j'}]$ or $s_i \,\Theta\, t_j$, where $\Theta \in \{<, >, \leq, \geq, =, \neq\}$. All the constraints appearing in Example 1.1 are expressible in $\mathrm{CDC}^{\neq}$. We show that for each constraint c in $\mathrm{CDC}^{\neq}$, a graph representation of c can be constructed in polynomial time. Let m be the number of time variables of one of the two local clocks, and n the other clock. The consistency of $c \in \mathrm{CDC}^{\neq}$ is decidable in $O(|c|mn)$ time, where $|c|$ is the size of c. Lastly, we show the intractability of constraints generated by disjointness constraints and conjunction and disjunction operators. For such general disjointness constraints, the consistency is NP-complete and the graph representability is coNP-complete.

The rest of the paper is organized as follows. Constraints on two clocks are formulated in section 2. In section 3, graph representations are introduced. It is also shown that the graph representability of a given constraint is coNP-complete. In section 4, a tractable subclass $\mathrm{CDC}^{\neq}$ of constraints is proposed. In section 5, the intractability of general disjointness constraints is shown. Section 6 compares the expressive powers of the classes of constraints. Section 7 summarizes the paper.

2. Constraints on two clocks. Let R be an infinite set of *global time points*. Suppose that a total order $\leq$ is defined on R. $r \leq r'$ means that point r precedes or is equal to r'. When $r \leq r'$ and $r \neq r'$, we write $r < r'$.

Let $S = \{s_0, s_1, \ldots, s_m\}$ and $T = \{t_0, t_1, \ldots, t_n\}$ $(m,\, n \geq 1)$ be sets of *variables*. We write $s_i \leq_S s_j$ and $t_i \leq_T t_j$ if $i \leq j$, and we write $s_i <_S s_j$ and $t_i <_T t_j$ if $i < j$. Intuitively, S and T are sets of *local time points*.

Let Σ_{ST} be the family of all the *valuations* $\sigma : S \cup T \to R$ satisfying the following conditions:

- $\sigma(s_0) = \sigma(t_0)$;
- $\sigma(s_m) = \sigma(t_n)$;
- if $s <_S s'$, then $\sigma(s) < \sigma(s')$;
- if $t <_T t'$, then $\sigma(t) < \sigma(t')$.

The first two conditions are introduced merely for theoretical simplicity. Namely, instead of the first condition, we can put dummy variables $s_{-\infty}$ and $t_{-\infty}$ such that $s_{-\infty} <_S s_0$, $t_{-\infty} <_T t_0$, and $\sigma(s_{-\infty}) = \sigma(t_{-\infty})$. On the other hand, the last two conditions are essential. σ must preserve the temporal orders of local time points.

Hereafter, we do not distinguish isomorphic valuations with respect to $<$ and $=$. In other words, we are interested in only the quotient sets of Σ_{ST} under $<$ and $=$. Therefore, σ will be regarded as a permutation of $S \cup T$ which is consistent with both $<_S$ and $<_T$ (although it may hold that $\sigma(s) = \sigma(t)$ for some $s \in S$ and $t \in T$), and Σ_{ST} will be regarded as the family of such permutations.

An *atomic constraint* is an expression with one of the following forms: $s < t$, $s > t$, $s \leq t$, $s \geq t$, $s = t$, and $s \neq t$. A *constraint* is generated from atomic constraints and Boolean operators $\neg$, $\vee$, and $\wedge$. For readability, we may use notation such as $t < s < t'$ to mean $(s > t) \wedge (s < t')$. The *satisfaction relation* is defined in an ordinary way, and we write $\sigma \models c$ (read as σ *satisfies* c) if c is true under valuation σ. If $\sigma \models c$ for some σ, then c is *consistent* (or *satisfiable*). By $c \models c'$, we mean that every valuation $\sigma \in \Sigma_{ST}$ satisfying c also satisfies c'. We say that c is *equivalent* to c', denoted $c \equiv c'$, if both $c \models c'$ and $c' \models c$ hold.

The *consistency problem* is to determine whether, given sets S and T of variables and constraint c, c is consistent or not. The *implication problem* is to determine whether $c \models c'$ holds or not for given sets S and T of variables and constraints c and c'. Since $c \models c'$ if and only if $\neg c \wedge c'$ is inconsistent, we mainly focus on the consistency problem in this paper.

Define the size of S and T as m and n, respectively. Also define the size of c as the number of atomic constraints in c.

THEOREM 2.1. *The consistency of an arbitrary constraint is in NP. The consistency of a constraint in conjunctive normal form (CNF) is NP-hard.*

Proof. The consistency problem is obviously in NP. The NP-hardness is shown by reducing the satisfiability problem of CNF logical formulas to this problem. In the reduction, each logical variable x_i in a given logical formula is replaced with an atomic constraint $s_i < t_i$. Whether $s_i < t_i$ holds or not can be determined independently of other atomic constraints $s_j < t_j$. Thus, the obtained constraint is consistent if and only if the original logical formula is satisfiable. ◻

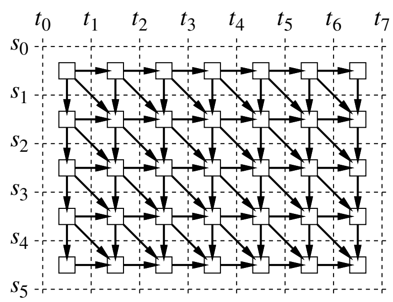


FIG. 2. G_{ST}.

3. Consistency and graph representability.

3.1. Graph representations of constraints. First, we define the graph G_{ST}.

DEFINITION 3.1. *Let $S = \{s_0, \ldots, s_m\}$ and $T = \{t_0, \ldots, t_n\}$. Define G_{ST} as a directed acyclic graph consisting of mn nodes arranged in m rows and n columns, with each node having outgoing arcs to the right, lower, and lower-right nodes (if existing). The node at the ith row of the jth column is denoted by (i, j).*

G_{ST} for $S = \{s_0, \ldots, s_5\}$ and $T = \{t_0, \ldots, t_7\}$ is shown in Figure 2. The $m + 1$ dotted horizontal lines (labeled $s_0, \ldots, s_m$) and $n + 1$ dotted vertical lines (labeled $t_0, \ldots, t_n$) are auxiliary lines explained below.

A *complete path* on G_{ST} is a path from $(1, 1)$ to (m, n). Let W_{ST} denote the set of all the complete paths on G_{ST}. There is a one-to-one correspondence between W_{ST} and Σ_{ST}. To see this, define a mapping $\rho : W_{ST} \to \Sigma_{ST}$ as follows. Let $w \in W_{ST}$.

- $\rho(w)(s_0) = \rho(w)(t_0)$ and $\rho(w)(s_m) = \rho(w)(t_n)$.
- If w crosses line s_i before line t_j, then $\rho(w)(s_i) < \rho(w)(t_j)$.
- If w crosses line t_j before line s_i, then $\rho(w)(s_i) > \rho(w)(t_j)$.
- If w crosses lines s_i and t_j at the same time, then $\rho(w)(s_i) = \rho(w)(t_j)$.

It can be shown that ρ is a bijection. Let ρ^{-1} denote the inverse of ρ.

If G is a subgraph of G', we write $G \subseteq G'$. The union $\cup$ (resp., intersection $\cap$) of subgraphs of G_{ST} is defined as the least upper bound (resp., greatest lower bound) of the subgraphs with respect to $\subseteq$.

The notion of complete paths is extended to subgraphs of G_{ST}. That is, for a subgraph G of G_{ST}, if there is a path from $(1, 1)$ to (m, n) on G, then the path is called a complete path on G.

DEFINITION 3.2. *Let c be a constraint and let G be a subgraph of G_{ST}. If the following two conditions hold, then G is a* graph representation *of c:*

- *$\rho(w) \models c$ for every complete path w on G; and*
- *$\rho^{-1}(\sigma)$ is a complete path on G for every valuation σ such that $\sigma \models c$.*

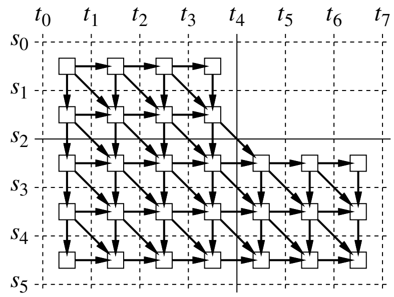


FIG. 3. *Minimum graph representation of $s_2 \leq t_4$.*

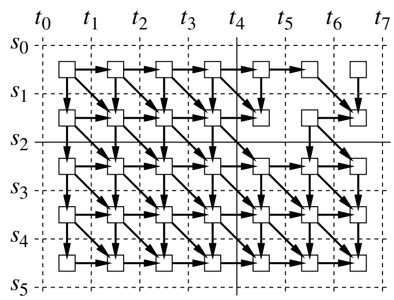


FIG. 4. *Another graph representation of $s_2 \leq t_4$.*

If c has a graph representation, then c is graph representable.

Some constraints have more than one graph representation, while some constraints have no graph representation. For example, both Figures 3 and 4 are graph representations of $s_2 \leq t_4$. On the other hand, as will be shown later, $(s_2 \leq t_4) \vee (s_4 \leq t_6)$ has no graph representation. Note that false has a graph representation (e.g., the empty graph).

For a graph-representable constraint c, let G_c denote an arbitrary graph representation of c.

LEMMA 3.3. *Let c and c' be graph-representable constraints. Then $G_c \cap G_{c'}$ is a graph representation of $c \wedge c'$.*

Proof. Any complete path w on $G_c \cap G_{c'}$ is contained in both G_c and $G_{c'}$. Therefore $\rho(w) \models c \wedge c'$. Conversely, consider any valuation σ such that $\sigma \models c \wedge c'$. Then $\rho^{-1}(\sigma)$ must be contained in both G_c and $G_{c'}$. Therefore, $\rho^{-1}(\sigma)$ is contained in $G_c \cap G_{c'}$. ◻

Suppose that c has a graph representation G_c. By Definition 3.2, c is consistent if and only if (m, n) is reachable from $(1, 1)$ on G_c. Since the reachability is decidable in $O(mn)$ time, we obtain the following lemma.

LEMMA 3.4. *Let $c = c_1 \vee \cdots \vee c_l$ be a disjunction of graph-representable constraints $c_1, \ldots, c_l$. Suppose that each G_{c_i} can be constructed in $O(f(c_i, m, n))$ time. Then the consistency of c is decidable in $O(f(c_1, m, n) + \cdots + f(c_l, m, n) + lmn)$ time.*

Proof. c is consistent if and only if some c_i is consistent. For each i, the consistency of c_i is decidable in $O(f(c_i, m, n) + mn)$ time. Therefore, the consistency of c is decidable in $O(f(c_1, m, n) + \cdots + f(c_l, m, n) + lmn)$ time. ◻

3.2. Minimum graph representations. A graph representation may contain redundant nodes and arcs. The following lemma states the existence of the simplest graph representation.

LEMMA 3.5. *Among all the graph representations of c, there is a unique minimum graph representation G_c^* with respect to $\subseteq$.*

Proof. The number of all the graph representations of c is finite since G_{ST} is a finite graph. Therefore, the intersection of all the graph representations of c can be defined. We show that the intersection G_c^* is the unique minimum graph representation of c. By Lemma 3.3, G_c^* is a graph representation of c $(= c \wedge c \wedge \cdots \wedge c)$. Clearly, $G_c^* \subseteq G_c$ for any graph representation G_c of c. Hence, G_c^* satisfies the lemma. ☐

LEMMA 3.6. *Let c be a graph-representable constraint. The minimum graph representation G_c^* of c can be constructed in $O(f(c,m,n)+mn)$ time, where $f(c,m,n)$ is the time complexity of constructing some graph representation of c.*

Proof. Suppose that a graph representation G_c of c is obtained. Then G_c^* can be constructed by the following algorithm:

1. Mark all the nodes and arcs that are reachable from $(1,1)$ in G_c. This can be done in $O(mn)$ time by performing depth first search from $(1,1)$.
2. Mark all the nodes and arcs from which (m,n) is reachable in G_c. This can be done in $O(mn)$ time by performing depth first search from (m,n), exploring the arcs in the opposite direction.
3. Excepting the nodes and arcs which are marked both in steps 1 and 2 above, remove the other nodes and arcs.

This algorithm is correct since the resultant graph contains only the nodes and arcs that are contained by a complete path on G_c. ☐

By Lemma 3.3, $G_c \cap G_{c'}$ is a graph representation of $c \wedge c'$. However, $G_c^* \cap G_{c'}^*$ is not necessarily the minimum graph representation of $c \wedge c'$. For example, consider G_c^* and $G_{c'}^*$ which have no common complete paths. Then $G_{c\wedge c'}^*$ is the empty graph, but $G_c^* \cap G_{c'}^*$ is not necessarily empty.

3.3. Complexity of deciding graph representability. We are interested in the complexity $f(c,m,n)$ of constructing G_c. Unfortunately, the graph representability of a given c with disjunction is coNP-complete (and therefore constructing G_c is coNP-hard), even if c is in disjunctive normal form (DNF). First, we characterize the graph representability.

LEMMA 3.7. *The following properties are equivalent:*

1. *c has no graph representation.*
2. *There is a complete path w on G_{ST} such that*
   - *$\rho(w) \not\models c$, and*
   - *for every arc a contained in w, there is another complete path w_a on G_{ST} containing the arc a such that $\rho(w_a) \models c$.*

Proof. Suppose that c has no graph representation. Let G be the minimum graph that contains all the complete paths satisfying c. Since c has no graph representation, there is a complete path w on G such that $\rho(w) \not\models c$. By the definition of G, for every arc a contained in w, there is another complete path w_a containing the arc a such that $\rho(w_a) \models c$. Thus the second property holds.

Conversely, suppose that the second property holds. Also assume that c has a graph representation G_c. Then G_c does not contain w of the second property, since G_c contains only the complete paths w' such that $\rho(w') \models c$. However, by the second property, for every arc a in w, there is w_a containing the arc a such that $\rho(w_a) \models c$. By definition, G_c contains w_a. This implies that all the arcs in w must be contained

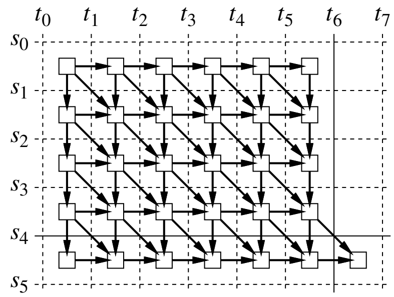


FIG. 5. *Minimum graph representation of* $s_4 \leq t_6$.

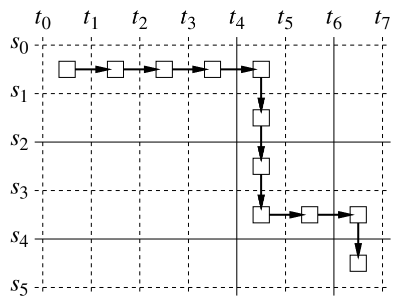


FIG. 6. *A complete path not satisfying* $(s_2 \leq t_4) \vee (s_4 \leq t_6)$.

in G_c, and contradicts the assumption that w is not a complete path on G_c. Thus, the second property implies the first one. □

We show that $c = (s_2 \leq t_4) \vee (s_4 \leq t_6)$ has no graph representation. The minimum graph representation of $s_4 \leq t_6$ is shown in Figure 5. Consider the complete path w shown in Figure 6. w does not satisfy c since w crosses line t_4 before line s_2, and crosses line t_6 before line s_4. However, for each arc a in w, there is another complete path w_a containing a such that $\rho(w_a) \models c$ (see Figures 3 and 5). By Lemma 3.7, c has no graph representation.

Before showing the coNP-completeness of the graph representability, we introduce a coNP-complete problem. Let F be a logical formula in DNF such that both ν_{true} and ν_{false} satisfy F, where ν_{true} (resp., ν_{false}) is the interpretation that maps every logical variable to true (resp., false). The *modified tautology problem* is to decide whether such a given logical formula F is a tautology (i.e., F is satisfied by all the interpretations). It is easily shown that the modified tautology problem is coNP-complete.

THEOREM 3.8. *The graph representability of an arbitrary constraint is in coNP. The graph representability of a constraint c with disjunction is coNP-hard, even if c is in DNF.*

Proof. The second property of Lemma 3.7 is decidable by an NP algorithm as follows. First, guess a complete path w and verify that $\rho(w) \not\models c$. Then, for every arc a contained in w, guess a complete path w_a containing a and verify that $\rho(w_a) \models c$.

To see the coNP-hardness, we reduce the modified tautology problem to the graph representability problem. Let $F = F_1 \vee \cdots \vee F_k$ be a DNF formula with n variables $x_1, \ldots, x_n$ such that both ν_{true} and ν_{false} satisfy F. Let

$$S = \{s_0, \ldots, s_{n+1}\}, \quad T = \{t_0, \ldots, t_{n+1}\}.$$

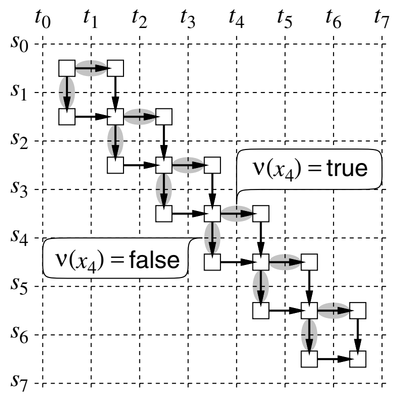


FIG. 7. *Graph representation of c' ($n = 6$).*

Let F'_l be the constraint obtained by replacing x_i in F_l by $s_i > t_i$, and $\neg x_i$ by $s_i < t_i$. Define $c_l = F'_l \wedge c'$, where

$$c' = \bigwedge_{i=1}^{n}((t_{i-1} < s_i < t_{i+1}) \wedge (s_i \neq t_i)).$$

The minimum graph representation $G^*_{c'}$ of c' is shown in Figure 7. Lastly, define $c = c_1 \vee \cdots \vee c_k$. Note that only the complete paths on $G^*_{c'}$ can satisfy c.

First, we show that F is a tautology if and only if all the complete paths on $G^*_{c'}$ satisfy c. With each interpretation ν of F, associate the following complete path w_ν on $G^*_{c'}$ (see Figure 7 again):

- w_ν contains $((i,i),(i,i+1))$ if $\nu(x_i) = \text{true}$;
- w_ν contains $((i,i),(i+1,i))$ if $\nu(x_i) = \text{false}$.

It is not difficult to see that ν satisfies F if and only if $\rho(w_\nu) \models c$. Note that $\rho(w_{\nu_{\text{true}}}) \models c$ and $\rho(w_{\nu_{\text{false}}}) \models c$ since both ν_{true} and ν_{false} satisfy F.

To complete the proof, we show that all the complete paths on $G^*_{c'}$ satisfy c if and only if c has a graph representation. For the *only if* part, suppose that all the complete paths on $G^*_{c'}$ satisfy c. Then, immediately from Definition 3.2, $G^*_{c'}$ is a graph representation of c. For the *if* part, suppose that $\rho(w) \not\models c$, where w is a complete path on $G^*_{c'}$. Then, for every arc a contained in w, there is another complete path w_a (namely, $w_{\nu_{\text{true}}}$ or $w_{\nu_{\text{false}}}$; see Figure 8) that contains a and $\rho(w_a) \models c$. Therefore, by Lemma 3.7, c has no graph representation. □

4. A tractable subclass of graph-representable constraints. We define a subclass CDC$^{\neq}$ of constraints such that for each constraint c in CDC$^{\neq}$, a graph representation of c can be constructed in $O(|c|mn)$ time.

DEFINITION 4.1. *A constraint in* CDC$^{\neq}$ *is a conjunction such that each conjunct is in the form of either $s \neq t$ or $(s \geq t') \vee (s' \leq t)$, where $s <_S s'$ and $t <_T t'$.*

CDC$^{\neq}$ can express any constraints in the form of $s_i \,\Theta\, t_j$, where $\Theta \in \{<, >, \leq, \geq, =, \neq\}$. For example,

$$(s_i \leq t_j) \equiv (s_0 \geq t_n) \vee (s_i \leq t_j),$$
$$(s_i < t_j) \equiv ((s_0 \geq t_n) \vee (s_i \leq t_j)) \wedge (s_i \neq t_j).$$

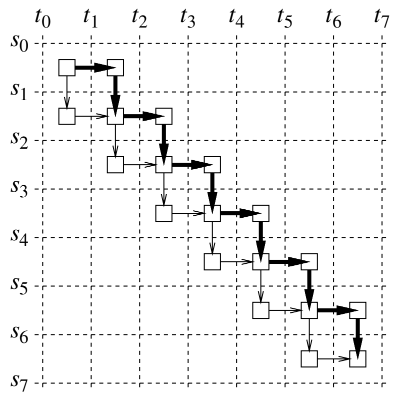


FIG. 8. $w_{\nu_{\text{true}}}$ *(thick arcs) and* $w_{\nu_{\text{false}}}$ *(thin arcs) in Theorem* 3.8 *(*$n = 6$*).*

Also, constraints in the form of $(s > t') \vee (s' \leq t)$, $(s \geq t') \vee (s' < t)$, or $(s > t') \vee (s' < t)$ are expressible in CDC$^{\neq}$. For example,

$$(s > t') \vee (s' \leq t) \equiv ((s \geq t') \vee (s' \leq t)) \wedge (s \neq t').$$

As stated in section 1, constraint $(s \geq t') \vee (s' \leq t)$ ($s <_S s'$ and $t <_T t'$) represents that time intervals $[s, s']$ and $[t, t']$ are disjoint. Therefore, we call the constraint a *disjointness constraint*, and we write $[s, s'] \not\pitchfork [t, t']$ to mean $(s \geq t') \vee (s' \leq t)$. The class of conjunctive disjointness constraints is denoted by CDC.

Now we show that constructing a graph representation of every constraint in CDC$^{\neq}$ is tractable. By Lemma 3.3, it suffices to show that each of the two forms in Definition 4.1 has a graph representation.

LEMMA 4.2. *The following two properties hold:*

1. *Let* $c = (s_i \neq t_j)$. *A graph representation of* c *is obtained by removing the arc* $((i, j), (i + 1, j + 1))$ *from* G_{ST}. *See Figure* 9*, for example.*
2. *Let* $c = ([s_i, s_{i'}] \not\pitchfork [t_j, t_{j'}])$. *Let* $N = \{(i'', j'') \mid i + 1 \leq i'' \leq i'$ *and* $j + 1 \leq j'' \leq j'\}$. *A graph representation of* c *is obtained by removing* N *and the adjacent arcs from* G_{ST}. *See Figure* 10*, for example.*

Proof. Since the first property is obvious, we consider only the case that $c = ([s_i, s_{i'}] \not\pitchfork [t_j, t_{j'}])$. Let G be the graph obtained by this lemma. We show that G is a graph representation of c. That is, $\sigma \models c$ if and only if $\rho^{-1}(\sigma)$ is a complete path on G.

Suppose that $\sigma \not\models c$. Then $\sigma \models (s_i < t_{j'}) \wedge (s_{i'} > t_j)$. This means that $\rho^{-1}(\sigma)$ contains some (i'', j'') such that $i + 1 \leq i'' \leq i'$ and $j + 1 \leq j'' \leq j'$. Since G does not contain (i'', j'') by definition, $\rho^{-1}(\sigma)$ is not a complete path on G.

Conversely, consider a complete path $\rho^{-1}(\sigma)$ on G_{ST} which is not contained in G. Then there is a node (i'', j'') ($i + 1 \leq i'' \leq i'$ and $j + 1 \leq j'' \leq j'$) in $\rho^{-1}(\sigma)$ which is contained in G_{ST} but not in G. Therefore, we conclude that $\sigma \models (s_i < t_{j'}) \wedge (s_{i'} > t_j)$. That is, $\sigma \not\models c$. ∎

THEOREM 4.3. *Let* $c \in$ CDC$^{\neq}$. *A graph representation of* c *can be constructed in* $O(|c|mn)$ *time. The minimum graph representation can also be constructed in* $O(|c|mn)$ *time.*

Proof. The proof is immediate from Lemmas 3.3, 3.6, and 4.2. ∎

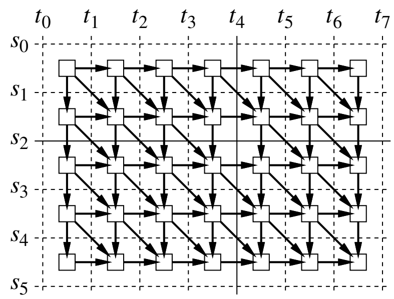


FIG. 9. *Minimum graph representation of* $s_2 \neq t_4$.

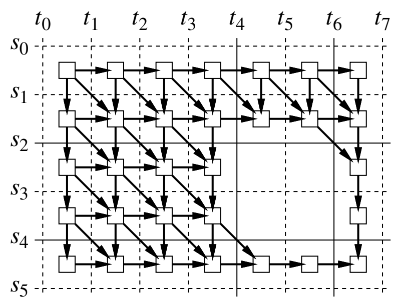


FIG. 10. *Minimum graph representation of* $[s_2, s_4] \not\cap [t_4, t_6]$.

THEOREM 4.4. *Let c be a disjunction of constraints in* $\mathrm{CDC}^{\neq}$. *The consistency of c is decidable in $O(|c|mn)$ time.*

Proof. The proof is immediate from Theorem 4.3 and Lemma 3.4. □

5. Intractability of general disjointness constraints. Recall the explanation of disjointness constraints in Example 1.1. Now we consider the case in which observations may contain uncertainty. For example, suppose that a_S observed that p or p' was true during time interval $[s, s']$. Also suppose that a_T observed that p was false during $[t, t']$, and p' was false during $[u, u']$. Then we obtain $([s, s'] \not\cap [t, t']) \vee ([s, s'] \not\cap [u, u'])$.

As stated above, observations with uncertainty bring disjunction into disjointness constraints. In this section, we show that both the consistency and the graph representability for disjointness constraints with disjunction are intractable.

5.1. Consistency of general disjointness constraints. We show that the consistency of general disjointness constraints is NP-complete.

THEOREM 5.1. *Let d be a constraint in CNF with respect to disjointness constraints; i.e., d is in the form of $d_1 \wedge \cdots \wedge d_n$, where each d_i is a disjunction of disjointness constraints. Then the consistency of d is NP-complete.*

Proof. By Theorem 2.1, the problem is in NP. To see the NP-hardness, we reduce the satisfiability of CNF logical formulas to the consistency problem.

Let F be a CNF formula with n variables $x_1, \ldots, x_n$. Let

$$S = \{s_0, \ldots, s_{2n+1}\}, \quad T = \{t_0, \ldots, t_{2n+1}\},$$

and let F' be the constraint obtained by replacing x_i in F by $[s_{2i}, s_{2i+1}] \not\cap [t_{2i-1}, t_{2i}]$,

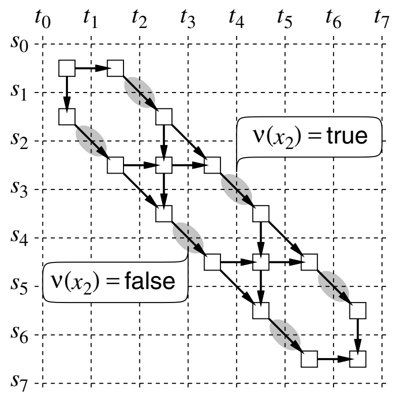


FIG. 11. *Graph representation of d' ($n = 3$).*

and $\neg x_i$ by $[s_{2i-1}, s_{2i}] \not\pitchfork [t_{2i}, t_{2i+1}]$. Also define

$$d' = \bigwedge_{i=1}^{n} ([s_{2i-1}, s_{2i}] \not\pitchfork [t_{2i-1}, t_{2i}]) \wedge \bigwedge_{j=1}^{2n-1} ([s_{j-1}, s_j] \not\pitchfork [t_{j+1}, t_{j+2}]) \wedge \bigwedge_{j=1}^{2n-1} ([s_{j+1}, s_{j+2}] \not\pitchfork [t_{j-1}, t_j]).$$

The minimum graph representation $G^*_{d'}$ of d' is shown in Figure 11. Last, define $d = F' \wedge d'$. Note that only the complete paths on $G^*_{d'}$ can satisfy d.

We show that d is consistent if and only if F is satisfiable. With each interpretation ν of F, associate a complete path w_ν on $G^*_{d'}$ satisfying the following conditions (see Figure 11 again):

- w_ν contains $((2i-1, 2i), (2i, 2i+1))$ if $\nu(x_i) = \text{true}$; and
- w_ν contains $((2i, 2i-1), (2i+1, 2i))$ if $\nu(x_i) = \text{false}$.

Such w_ν always exists. On the other hand, for every complete path w on $G^*_{d'}$, there is ν such that $w = w_\nu$.

Suppose that $\nu(x_i) = \text{true}$. Then w_ν contains $((2i-1, 2i), (2i, 2i+1))$, and therefore $\rho(w_\nu)(s_{2i-1}) = \rho(w_\nu)(t_{2i})$. This means that $\rho(w_\nu)$ satisfies $[s_{2i}, s_{2i+1}] \not\pitchfork [t_{2i-1}, t_{2i}]$ but not $[s_{2i-1}, s_{2i}] \not\pitchfork [t_{2i}, t_{2i+1}]$. Similarly, suppose that $\nu(x_i) = \text{false}$. Then w_ν contains $((2i, 2i-1), (2i+1, 2i))$, and therefore $\rho(w_\nu)(s_{2i}) = \rho(w_\nu)(t_{2i-1})$. This means that $\rho(w_\nu)$ satisfies $[s_{2i-1}, s_{2i}] \not\pitchfork [t_{2i}, t_{2i+1}]$ but not $[s_{2i}, s_{2i+1}] \not\pitchfork [t_{2i-1}, t_{2i}]$. Thus ν satisfies F if and only if $\rho(w_\nu) \models d$. $\square$

5.2. Graph representability of general disjointness constraints. We show that the graph representability of general disjointness constraints is coNP-complete.

THEOREM 5.2. *Let d be a constraint in DNF with respect to disjointness constraints; i.e., d is in the form of $d_1 \vee \cdots \vee d_n$, where each d_i is a conjunction of disjointness constraints. Then the graph representability of d is coNP-complete.*

Proof. By Theorem 3.8, the problem is in coNP. To see the coNP-hardness, we reduce the modified tautology problem to the graph representability problem.

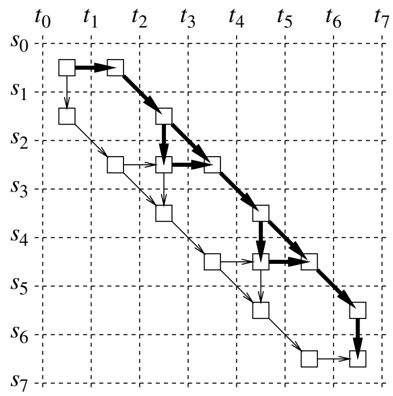


FIG. 12. $w_{\nu_{\text{true}}}$ *(thick arcs) and* $w_{\nu_{\text{false}}}$ *(thin arcs) in Theorem* 5.2 *(*$n = 3$*).*

Let $F = F_1 \vee \cdots \vee F_k$ be a DNF formula with n variables $x_1, \ldots, x_n$ such that both ν_{true} and ν_{false} satisfy F. Let

$$S = \{s_0, \ldots, s_{2n+1}\}, \quad T = \{t_0, \ldots, t_{2n+1}\}.$$

Let F'_l be the constraint obtained by replacing x_i in F_l by $[s_{2i}, s_{2i+1}] \not\pitchfork [t_{2i-1}, t_{2i}]$, and $\neg x_i$ by $[s_{2i-1}, s_{2i}] \not\pitchfork [t_{2i}, t_{2i+1}]$. Then define $d_l = F'_l \wedge d'$, where d' is the same as the proof of Theorem 5.1. Last, define $d = d_1 \vee \cdots \vee d_k$.

In the same way as Theorem 5.1, associate a complete path w_ν on $G^*_{d'}$ with each interpretation ν of F. Then ν satisfies F if and only if $\rho(w_\nu) \models d$. Therefore, F is a tautology if and only if all the complete paths on $G^*_{d'}$ satisfy d. Note that $\rho(w_{\nu_{\text{true}}}) \models d$ and $\rho(w_{\nu_{\text{false}}}) \models d$ since both ν_{true} and ν_{false} satisfy F.

To complete the proof, we show that all the complete paths on $G^*_{d'}$ satisfy d if and only if d has a graph representation. For the *only if* part, suppose that all the complete paths on $G^*_{d'}$ satisfy d. Then, immediately from Definition 3.2, $G^*_{d'}$ is a graph representation of d. For the *if* part, suppose that $\rho(w) \not\models d$, where w is a complete path on $G^*_{d'}$. Then, for every arc a contained in w, there is another complete path w_a (namely, $w_{\nu_{\text{true}}}$ or $w_{\nu_{\text{false}}}$; see Figure 12) that contains a and $\rho(w_a) \models d$. Therefore, by Lemma 3.7, d has no graph representation. $\square$

6. Comparison to the related works.

6.1. Related works. Temporal reasoning has been extensively studied mainly in the field of artificial intelligence. Allen [2] proposed the *interval algebra*, which can express the conjunction of any relation between two time intervals. Table 1 shows the 13 basic operators of the interval algebra. Every relation between two time intervals is represented by a disjunctive combination of some of these basic operators. For example, the disjointness of two intervals I and J is represented by $I(\mathsf{pp}^{-1}\mathsf{mm}^{-1})J$, i.e., either I precedes J, I is preceded by J, I meets J, or I is met by J. Unfortunately, many of the basic problems including the consistency (i.e., satisfiability) for the interval algebra are NP-hard [12]. Therefore, most of the researches aim to find tractable classes of temporal constraints.

One of the research directions is to weaken the expressive power of the interval algebra. Vilain and Kautz [12] proposed the *point algebra*, which can express the conjunction of any relationship between two time points. The point algebra is strictly

TABLE 1
Basic interval-interval operators.

I precedes J	p	`III`
J preceded by I	p^{-1}	`         JJJ`
I meets J	m	`IIII`
J met by I	m^{-1}	`       JJJJ`
I overlaps J	o	`IIII`
J overlapped by I	o^{-1}	`   JJJJ`
I during J	d	`   III`
J includes by I	d^{-1}	`JJJJJJJ`
I starts J	s	`III`
J started by I	s^{-1}	`JJJJJJJ`
I finishes J	f	`    III`
J finished by I	f^{-1}	`JJJJJJJ`
I equals J	$\equiv$	`IIII` `JJJJ`

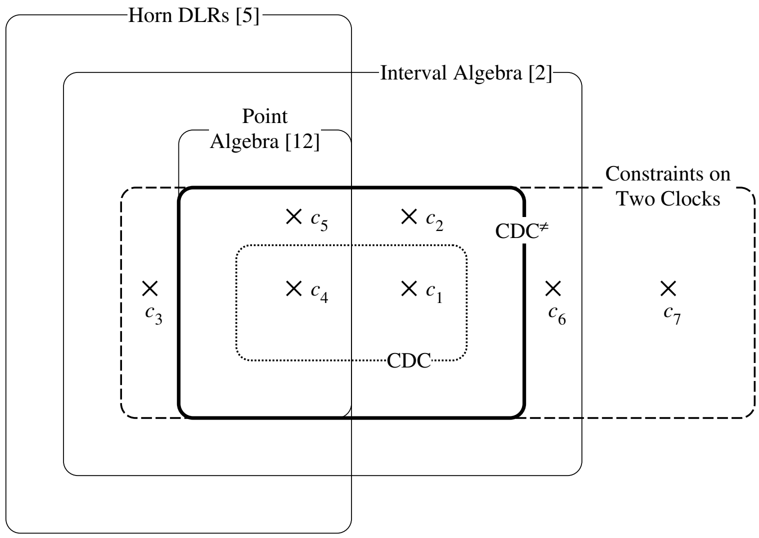
less expressive than the interval algebra, and the consistency of a given constraint c is decidable in $O(|c|^2)$ time [13]. Golumbic and Shamir [3] investigated several subalgebras of the interval algebra for which the consistency is decidable in polynomial time. Nebel and Bürckert [9] proposed a subalgebra of the interval algebra called ORD-Horn, which contains the point algebra. They showed that ORD-Horn is a maximal tractable subalgebra. More precisely, the consistency of a given ORD-Horn constraint c is decidable in $O(|c|^3)$ time, while the consistency for any subalgebra of the interval algebra which strictly contains ORD-Horn is NP-complete. Recently, all the tractable subalgebras of the interval algebra were discovered by Krokhin, Jeavons, and Jonsson [6].

On the other hand, some researches focus on classes incomparable to the interval algebra. Jonsson and Bäckström [5] proposed a class of temporal constraints called Horn DLRs. The class of Horn DLRs is a superclass of ORD-Horn and includes quantitative constraints (and therefore Horn DLRs are incomparable to the interval algebra). The consistency of Horn DLRs is decidable in polynomial time using a fast algorithm for the linear programming problem. Van der Meyden [11] studied the complexity of determining whether a conjunction of given constraints in the form of $s \leq t$ or $s' < t$ implies a given negation-free existentially quantified constraint.

6.2. Comparison of the expressive power. The relationship among classes of constraints is shown in Figure 13.

First of all, note that ORD-Horn and CDC$^{\neq}$ are incomparable. In a constraint of ORD-Horn, every relation between intervals must be expressed by a conjunction of *ORD-Horn clauses*. An ORD-Horn clause is a disjunction of at most one positive atomic constraint (i.e., $s \leq t$ or $s = t$) and an arbitrary number of negative atomic constraints (i.e., $s \neq t$). Since a disjointness constraint $c_1 = [s_1, s_2] \not\cap [t_1, t_2]$ is a disjunction of two positive atomic constraints, it is not in ORD-Horn. Also, $c_2 = (s_1 \neq t_1) \wedge ([s_1, s_2] \not\cap [t_1, t_2])$ is in CDC$^{\neq}$, but in neither CDC nor ORD-Horn. On the other hand, ORD-Horn includes constraints that are not in CDC$^{\neq}$. For example, $c_3 = (s_1 \neq t_1) \vee (s_2 \neq t_2)$ is in ORD-Horn but not in CDC$^{\neq}$. Also, ORD-Horn includes some of the constraints on more than two clocks, but CDC$^{\neq}$ never includes them.

Next, note that CDC$^{\neq}$ $\cap$ "ORD-Horn" = CDC$^{\neq}$ $\cap$ "point algebra". Any clause of a constraint in CDC$^{\neq}$ consists of one atomic constraint or two positive atomic constraints. Therefore, any clause of a constraint in CDC$^{\neq}$ $\cap$ "ORD-Horn" consists



"ORD-Horn" [9] = "Horn DLRs" ∩ "Interval Algebra"

FIG. 13. *Comparison of the expressive power of the classes of constraints.*

of one atomic constraint, and hence is expressible in the point algebra.

Class CDC and the point algebra have a nonempty intersection since it contains $c_4 = ([s_0, s_i] \not\pitchfork [t_j, t_n]) \equiv (s_i \leq t_j)$ as stated in section 4. On the other hand, $c_5 = (s_2 \neq t_4)$ is not in CDC because the minimum graph representation of c_5 (Figure 9) cannot be a subgraph of an intersection of graph representations of disjointness constraints (Figure 10).

Let $c_6 = ([s_0, s_2] \not\pitchfork [t_4, t_5]) \vee ([s_0, s_4] \not\pitchfork [t_6, t_7])$. Then

$$\begin{aligned} c_6 &\equiv (s_0 \geq t_5) \vee (s_2 \leq t_4) \vee (s_0 \geq t_7) \vee (s_4 \leq t_6) \\ &\equiv (s_2 \leq t_4) \vee (s_4 \leq t_6). \end{aligned}$$

This is not in CDC$^{\neq}$ because c_6 has no graph representation as shown in section 3. On the other hand, c_6 is in the interval algebra, i.e., $c_6 \equiv [s_2, s_4](\mathsf{pmodd}^{-1}\mathsf{ss}^{-1}\mathsf{ff}^{-1}\equiv)[t_4, t_6]$.

Last, $c_7 = ([s_1, s_2] \not\pitchfork [t_1, t_2]) \vee ([s_3, s_4] \not\pitchfork [t_3, t_4])$ is a constraint on two clocks but not in the interval algebra, because in the interval algebra, disjunction of relations of distinct pairs of intervals is not expressible.

After the first submission of this paper, all the maximal tractable subalgebras of the interval algebra were identified by Krokhin, Jeavons, and Jonsson [6]. CDC$^{\neq}$ is incomparable to any of the tractable subalgebras. The outline of the proof is as follows. Let A be a subalgebra of the interval algebra that can express all the constraints in CDC$^{\neq}$. A must contain $(\mathsf{pp}^{-1}\mathsf{mm}^{-1})$ in order to express $\not\pitchfork$. Also A must contain (m) or (m^{-1}) in order to express adjacency of intervals (e.g., $[s_0, s_1]$ and $[s_1, s_2]$ in Example 1.1). Then, from the definitions of the discovered tractable subalgebras (Table 3 of [6]), it is immediate that A is not contained in any of the tractable subalgebras.

Recently, Krokhin and Jonsson [7] also found maximal tractable subclasses of Meiri's qualitative algebra [8]. The qualitative algebra contains all the relational

TABLE 2
Basic point-interval operators.

p before I	`b`	`p` `   III`
p starts I	`s`	`p` `III`
p during I	`d`	` p` `III`
p finishes I	`f`	`  p` `III`
p after I	`a`	`     p` `III`

operators between points, between intervals, and between a point and an interval (see Table 2). CDC$^{\neq}$ is also incomparable with any of the known tractable subclasses of the qualitative algebra. The outline of the proof is as follows. Let A' be a subclass of the qualitative algebra that can express all the constraints in CDC$^{\neq}$. First, it can be shown that $\not\cap$ is not expressible by using only point-point and point-interval operators. Therefore, A' must contain $(\mathsf{pp}^{-1}\mathsf{mm}^{-1})$. Next, we can conclude that in order to express adjacency of intervals, either (1) A' must contain either (m) or (m^{-1}); or (2) A' must contain both $(\mathtt{s})$ and $(\mathtt{f})$. Then it is not difficult to see that A' is not contained in any of the known tractable subclasses.

7. Conclusions. In this paper, we have studied temporal reasoning with respect to constraints on two concurrent sequences of events. We have introduced the notion of graph representations of constraints. If a graph representation of a given constraint c can be constructed in polynomial time, then the consistency of c is also decidable in polynomial time. We have proposed a subclass CDC$^{\neq}$ of constraints such that a graph representation of a constraint in CDC$^{\neq}$ can be constructed in polynomial time.

We have considered only the case in which the number of local clocks is *two*. However, this assumption is for simplicity. If the number of local clocks is an *arbitrary* constant, then the idea of graph representability is applicable and the consistency is decidable in polynomial time.

As future work, constraints on durations [10] and/or quantitative constraints on two concurrent sequences of events should be studied. First-order constraints on two concurrent sequences of events also remain to be investigated.

Acknowledgments. The authors would like to thank Mr. Tetsuya Murakami of Furukawa Electric Co., Ltd. for his helpful discussions. The authors are also grateful to the anonymous referees for their valuable suggestions and comments.

REFERENCES

[1] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*, Addison-Wesley, Boston, MA, 1995.
[2] J. F. Allen, *Maintaining knowledge about temporal intervals*, Comm. ACM, 26 (1983), pp. 832–843.
[3] M. C. Golumbic and R. Shamir, *Complexity and algorithms for reasoning about time: A graph-theoretic approach*, J. ACM, 40 (1993), pp. 1108–1133.
[4] H. Isozaki and H. Katsuno, *Observability-based nested belief computation for multiagent systems and its formalization*, in Proceedings of the 6th International Workshop on Agent Theories, Architectures, and Languages (ATAL'99), Lecture Notes in Artificial Intelligence 1757, Springer-Verlag, Berlin, 2000, pp. 27–41.

[5] P. Jonsson and C. Bäckström, *A unifying approach to temporal constraint reasoning*, Artificial Intelligence, 102 (1998), pp. 143–155.

[6] A. Krokhin, P. Jeavons, and P. Jonsson, *Reasoning about temporal relations: The tractable subalgebras of Allen's interval algebra*, J. ACM, 50 (2003), pp. 591–640.

[7] A. Krokhin and P. Jonsson, *Extending the point algebra into the qualitative algebra*, in Proceedings of the 9th International IEEE Symposium on Temporal Representation and Reasoning (TIME'02), 2002, pp. 28–35.

[8] I. Meiri, *Combining qualitative and quantitative constraints in temporal reasoning*, Artificial Intelligence, 87 (1996), pp. 343–385.

[9] B. Nebel and H.-J. Bürckert, *Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra*, J. ACM, 42 (1995), pp. 43–66.

[10] A. K. Pujari, G. V. Kumari, and A. Sattar, *INDU: An interval and duration network*, in Proceedings of the Australian Joint Conference on Artificial Intelligence, Lecture Notes in Comput. Sci. 1747, Springer-Verlag, Berlin, 1999, pp. 291–303.

[11] R. van der Meyden, *The complexity of querying indefinite data about linearly ordered domains*, J. Comput. System Sci., 54 (1997), pp. 113–135.

[12] M. B. Vilain and H. A. Kautz, *Constraint propagation algorithms for temporal reasoning*, in Proceedings of the 5th National AAAI Conference on Artificial Intelligence (AAAI'86), 1986, pp. 377–382.

[13] M. B. Vilain, H. A. Kautz, and P. van Beek, *Constraint propagation algorithms for temporal reasoning: A revised report*, in Readings in Qualitative Reasoning about Physical Systems, Morgan-Kaufmann, San Mateo, CA, 1990, pp. 373–381.