# Towards a Complete Classification of Tractability in Point Algebras for Nonlinear Time

Mathias Broxvall * and Peter Jonsson **

Department of Computer and Information Science
Linköpings Universitet
S-581 83 Linköping, Sweden
{matbr,petej}@ida.liu.se

**Abstract.** Efficient reasoning about temporal constraints over nonlinear time models is vital in numerous application areas, such as planning, distributed systems and cooperating agents. We identify all tractable subclasses of the point algebra for partially-ordered time and examine one large, nontrivial tractable subclass of the point algebra for branching time.

## 1 Introduction

Reasoning about temporal knowledge is a common task in computer science and elsewhere. In most applications, knowledge of temporal constraints is expressed in terms of collections of relations between time intervals or time points. Typical reasoning tasks include determining the satisfiability of such collections and deducing new relations from those that are known.

Research on automated reasoning about temporal constraints has largely concentrated on simple, linear models of time. However, it has been observed over and over again that more complex time models are needed in the analysis of concurrent and distributed systems, where simultaneous actions and their interactions must be represented and the order relationships between events are recorded by unsynchronized clocks. Typical problems in distributed system development include the traditional difficulties of synchronization and communication problems, network and bus protocol correctness and decision making in the absence of a global clock. Nonlinear time models are also needed in a number of other applications such as certain planning domains, robot motion problems and cooperating agents, to mention only a few.

For analyzing such systems, a number of time models have been proposed. Examples include temporal logics corresponding to *branching* time [6, 4] and *partially-ordered* time [1, 9]. The branching time model has been especially successful in its application to planning problems [5, 10]. Since most of the previous

* This research has been supported by the ECSEL graduate student program.

** This research has been supported by the Swedish Research Council for the Engineering Sciences (TFR) under grant 97-301.

research on nonlinear time have concentrated on modelling properties and theoretical foundations, computational aspects have received relatively little attention. There are a few papers that address such questions, though. Clarke *et al.* [4] have provided an apparently efficient algorithm than can check the correctness of certain circuits and algorithms. The method builds on a propositional branching-time logic based on CTL (Computation Tree Logic). Anger *et al.* [2] have studied whether standard constraint-based approaches can be applied to reasoning about nonlinear time; their results show that this is possible in certain restricted cases.

The main result of this paper is a total classification of the point algebra over partially-ordered time (by partially-ordered time, we mean a time structure where the time points are ordered by some partial order). We show that there exists exactly three maximal tractable classes and our proof does not rely on computer-assisted case analyses which has been the standard method in solving similar problems (*cf.* [11, 8]). We also examine the point algebra for branching time and identify a large, nontrivial tractable subclass (by *branching time*, we mean a partially-ordered time structure where no two incomparable time points have an upper bound. Intuitively, this condition prohibits joining of different branches of time).

The paper is structured as follows: Section 2 contains basic definitions and some auxiliary results. Section 3 and Sect. 4 contains the results on partially-ordered and branching time, respectively. Some concluding remarks are given in Section 5. For a technical report containing somewhat more exhaustive proofs as well as examples see Broxvall [3].

## 2 Preliminaries

The point algebra is based on the notion of *relations* between pairs of variables interpreted over a partially-ordered set. In this paper we consider four *basic relations* which we denote by $<, >, =$ and $\|$. If $x, y$ are points in a partial order $\langle T, \leq \rangle$ then we define these relations in terms of the partial ordering $\leq$ as follows:

1. $x < y$ iff $x \leq y$ and not $y \leq x$
2. $x > y$ iff $y \leq x$ and not $x \leq y$
3. $x = y$ iff $x \leq y$ and $y \leq x$
4. $x \| y$ iff neither $x \leq y$ nor $y \leq x$

The relations we consider are always disjunctions of basic relations and they are represented as sets of basic relations. Since we have 4 different basic relations we get $2^4 = 16$ possible disjunctive relations. The set of basic relations is denoted $\mathcal{B}$ and the set of all 16 relations is denoted by $\mathcal{PA}$. Sometimes we use a short-hand notation for certain relations, for example, $\{<, =\}$ is sometimes written as $\leq$ and $\{=, \|\}$ as $\underline{\|}$. The empty relations is denoted by $\perp$ and it is always unsatisfied.

The basic computational problem of the point algebra is the satisfiability problem where we have a set of variables and a set of constraints over the variables and the question is whether there exists a mapping from the variables to a specific domain such that all constraints are satisfied. Here we choose to consider a problem instance to be a graph where the nodes represents the variables and the labelled edges represents the constraints.

**Definition 1.** *Let $\Re \subseteq \mathcal{PA}$ be a set of point relations and* P *a class of partial orders. A problem instance of $PSAT_{\mathrm{P}}(\Re)$ is a directed multigraph $\Pi = \langle V, E \rangle$ where the nodes in $V$ are point variables and $E \subseteq V \times \Re \times V$ denotes the constraints imposed on the variables. A tuple $\langle f, \langle T, \leq \rangle \rangle$ where $f : V \to T$ is a total function and $\langle T, \leq \rangle \in P$ is called an interpretation of $\Pi$.*

*A problem instance $\Pi$ is satisfiable iff there exists an interpretation $M = \langle f, \langle T, \leq \rangle \rangle$ such that $f(u)\ R\ f(v)$ holds for every $\langle u, R, v \rangle \in E$. $M$ is called a* model *of $\Pi$. Given a model $M = \langle f, \langle T, \Re \rangle \rangle$, we sometimes write $M(x)$ to denote $f(x)$.*

*The size of a problem instance $\langle V, E \rangle$ is $|V| + |E|$.*

Given an instance $\Pi$ of $\mathrm{PSAT}_{\mathrm{p}}(\Re)$ and two variables $x, y$ we write $x \leq^{+} y$ to say that there exists zero or more variables $z_1, \ldots, z_n$ such that

$$x \leq z_1 \wedge z_1 \leq z_2 \wedge \cdots \wedge z_{n-1} \leq z_n \wedge z_n \leq y$$

and we write $x \leq^{*} y$ to say $x \leq^{+} y$ or $x = y$. An $R$-subgraph $\Pi'$ of $\Pi$ is defined as $\langle V, E \cap (V \times \{R\} \times V) \rangle$. A node $n$ is *$R$-minimal* of the graph $\Pi$ iff $n$ is minimal in the $R$-subgraph of $\Pi$. The *$R$-components* of a graph $\Pi$ are the components of the $R$-subgraph of $\Pi$. A graph $\Pi$ is $(\leq)$-acyclic iff the $(\leq)$-subgraph of $\Pi$ is acyclic.

We will consider two classes of partial orders: *po* which is the class of all partial orders and $br \subset po$ which is the class of all partial orders satisfying the *branching condition*:

$$\forall x, y, z : \text{IF } x \| y \text{ AND } y < z \text{ THEN } x \| z.$$

Note that in a partial order of class *br* two incomparable points cannot have an upper bound.

Next, we define two operations on point relations. The *intersection* operator takes two point relations $R_1, R_2$ to their intersection $R_3 = R_1 \cap R_2$ such that: $\forall x, y : x\ R_3\ y \leftrightarrow x\ R_1\ y \wedge x\ R_2\ y$. The *composition* operator takes two point relations $R_1, R_2$ to their composition $R_3 = R_1 \circ R_2$ such that: $\forall x, y : x\ R_3\ y \leftrightarrow \exists z : x\ R_1\ z \wedge z\ R_2\ y$. The composition of $R_1, R_2$ evaluates to the union of the composition of each pair of basic relations $r_1 \in R_1, r_2 \in R_2$. Composition tables for *po* and *br* can be found in Table 1.

We will now introduce the concept of gadgets which play an imporant role since they allow us to concentrate on smaller sets of relations while proving tractability for a larger set which can be implemented in terms of the smaller set.

**Definition 2.** *Let $\Theta$ be an instance of $PSAT_{\mathrm{p}}(\Re)$ containing the variables $x, y, z_1, \cdots, z_n$. Assume that the following holds: In every model of $\Theta$, $x\ R\ y$ holds for some $R \in \mathcal{PA}$. Then we say that $\Re$* implements *the relation $R$ and we say that $\Theta$ is a* gadget *that implements $R$.*

**Table 1.** Composition of basic relations in partially-ordered time (left figure) and branching time (right figure).

| | $<$ | $>$ | $=$ | $\Vert$ |
|---|---|---|---|---|
| $<$ | $\{<\}$ | $\mathcal{B}$ | $\{<\}$ | $\{\Vert <\}$ |
| $>$ | $\mathcal{B}$ | $\{>\}$ | $\{>\}$ | $\{> \Vert\}$ |
| $=$ | $\{<\}$ | $\{>\}$ | $\{=\}$ | $\{\Vert\}$ |
| $\Vert$ | $\{\Vert <\}$ | $\{> \Vert\}$ | $\{\Vert\}$ | $\mathcal{B}$ |

| | $<$ | $>$ | $=$ | $\Vert$ |
|---|---|---|---|---|
| $<$ | $\{<\}$ | $\{<=>\}$ | $\{<\}$ | $\{\Vert <\}$ |
| $>$ | $\mathcal{B}$ | $\{>\}$ | $\{>\}$ | $\{\Vert\}$ |
| $=$ | $\{<\}$ | $\{>\}$ | $\{=\}$ | $\{\Vert\}$ |
| $\Vert$ | $\{\Vert\}$ | $\{> \Vert\}$ | $\{\Vert\}$ | $\mathcal{B}$ |

The intersection and composition operators are frequently used to verify that a given gadget implements a certain relation.

*Example 1.* The following gadget implements $\{<>\}$ for the branching point algebra since $\{<=\} \circ \{>=\}$ is $\{<=>\}$ and $\{<=>\} \cap \{< \Vert >\}$ is $\{<>\}$.

$$\begin{tikzcd} x \arrow[r, bend right, "\leq"'] \arrow[rr, bend left, leftrightarrow, "\neq"] & z \arrow[r, bend right, "\geq"'] & y \end{tikzcd}$$

Gadgets and implementations play an important role in proving tractability and NP-completeness results and the main vehicle is the following simple lemma.

**Lemma 1.** *If $\Re$ implements $R_1, \ldots, R_k$ and $PSAT_{\mathrm{p}}(\Re)$ is tractable, then $PSAT_{\mathrm{p}}(\Re \cup \{R_1, \ldots, R_k\})$ is tractable. If $\Re$ implements $R_1, \ldots, R_k$ and $PSAT_{\mathrm{p}}(\{R_1, \ldots, R_k\})$ is NP-hard, then $PSAT_{\mathrm{p}}(\Re)$ is NP-hard.*

*Proof.* Assume $\mathrm{PSAT_p}(\Re)$ to be solvable in polynomial time. We replace each edge marked with $R$ in a problem instance $\Pi = \langle V, E\rangle$ one by one with the edges and nodes in a gadget implementing $R$ until no more such edges exists. Let $\langle a, R, b\rangle \in E$ be an arbitrary edge marked with $R$ and let $\Theta = \langle\{a, b, v_1, \cdots, v_n\}, E'\rangle$ be a gadget that implements $a\ R\ b$. Construct

$$\Pi' = \langle V \cup \{v_1, \cdots, v_n\}, E \cup E' - \langle a, R, b\rangle\rangle$$

Clearly $\Pi'$ is satisfiable iff $\Pi$ is satisfiable. Repeat until no more edges marked with $R$ exists. Clearly all reductions can be performed in polynomial time. Hence, $\mathrm{PSAT_p}(\Re)$ is solvable in polynomial time. $\square$

If $f$ is a function which is undefined for $n$, then $(f|n \longmapsto c)$ is defined as the function $f \cup \{\langle n, c\rangle\}$. Let $n_1, n_2$ be nodes in a graph $\Pi$ and let,

$$\begin{aligned} \Pi' = \langle V - \{n_2\}, E &\cup \{\langle n_1, x\rangle | \langle n_2, x\rangle \in E\} \\ &\cup \{\langle x, n_1\rangle | \langle x, n_2\rangle \in E\} \\ &- \{\langle x, n_2\rangle, \langle n_2, x\rangle | x \in V\}\rangle \end{aligned}$$

We say that $\Pi'$ is obtained by *contracting* $n_1, n_2$. That is, we identify the nodes $n_1, n_2$ by $n_1$. Note that there may be edges from $n_1$ to $n_1$.

Finally, we give a simple result which is needed several times to prove correctness of different algorithms.

**Table 2.** The classes $\mathcal{A}_{14}$, $\mathcal{A}_{13}$, $\mathcal{A}_{9}$, and $\mathcal{A}_{10}$

| | $\mathcal{A}_{14}$ | $\mathcal{A}_{13}$ | $\mathcal{A}_{9}$ | $\mathcal{A}_{10}$ |
|---|---|---|---|---|
| $\perp$ | • | • | • | • |
| $\{<\}$ | • | • | | • |
| $\{>\}$ | • | • | | • |
| $\{<>\}$ | | • | | • |
| $\{\parallel\}$ | • | • | | |
| $\{\parallel <\}$ | • | • | | |
| $\{> \parallel\}$ | • | • | | |
| $\{< \parallel >\}$ | • | • | | • |
| $\{=\}$ | • | • | • | • |
| $\{<=\}$ | • | • | • | • |
| $\{>=\}$ | • | • | • | • |
| $\{<=>\}$ | | • | • | • |
| $\{= \parallel\}$ | • | | • | |
| $\{<= \parallel\}$ | • | | • | |
| $\{>= \parallel\}$ | • | | • | |
| $\mathcal{B}$ | • | • | • | • |

**Lemma 2.** *Let $\Pi$ be an instance of $PSAT_{\mathrm{p}}(\Re)$ such that for every model $M$ of $\Pi$ holds that $M(n_1) = M(n_2)$ and $\Pi'$ be the graph resulting from contracting the nodes $n_1, n_2$ in $\Pi$. Then, $\Pi$ is satisfiable iff $\Pi'$ is satisfiable.*

*Proof.* Assume $\Pi$ has a model $M$. Clearly $M$ is also a model of $\Pi'$. Assume $\Pi'$ has a model $M' = \langle f, \langle T, \Re\rangle\rangle$, the following is a model of $\Pi$:

$$M = \langle (f | n_2 \longmapsto f(n_1)), \langle T, \Re\rangle\rangle$$

□

# 3 Partially-Ordered Time

When examining the point algebra for partially-ordered time we begin by examining three classes $\mathcal{A}_{14}$, $\mathcal{A}_{10}$ and $\mathcal{A}_{9}$ and show that they are tractable. These results are collected in Subsection 3.1. We continue by showing that these classes are maximal[1] and that there exists no other maximal tractable classes in Subsection 3.2. The classes are defined in Table 2.

## 3.1 Tractable subclasses

To prove the tractability of $\mathcal{A}_{14}$ and $\mathcal{A}_{10}$ we present an algorithm PO in Fig. 3.1 and claim that it correctly solves the problems $\mathrm{PSAT}_{\mathrm{po}}(\mathcal{C})$ (where $\mathcal{C} = \{\leq, \underline{\parallel}, \neq\}$)

[1] We say that a set of relations $C$ is a *maximal tractable* subclass of $\mathcal{PA}$ iff $\mathrm{PSAT}_{\mathrm{p}}(C)$ is tractable and there exist no proper superclass of $C$ that is tractable.

```
algorithm PO
Input: An instance Π of PSAT_po(C) or PSAT_po(D)
 repeat
   Π' ← Π
   for each pair of nodes n1, n2 ∈ Π  do
     if n1||n2 and n1 ≤* n2  then
       if n1 ≠ n2  then  return false
       else  Π' ← contract(Π, n1, n2)
     elsif n1 ≤* n2 and n2 ≤* n1  then
       if n1 ≠ n2 ∈ Π  then  return false
       else  Π' ← contract(Π, n1, n2)
     end if
   end for
until Π' = Π
return true
```

**Fig. 1.** Algorithm PO

and $\mathrm{PSAT}_{\mathrm{po}}(\mathcal{D})$ (where $\mathcal{D} = \{\leq, \{<=>\}, \neq\}$). By proving that $\mathcal{C}$ implements $\mathcal{A}_{14}$ and $\mathcal{D}$ implements $\mathcal{A}_{10}$, we are done. Note that algorithm PO runs in polynomial time since every loop removes at least one element from the given instance $\Pi$, so at most $|\Pi|$ loops can be made. Algorithm PO return boolean values where the answer "true" means the algorithm accepts the given problem instance (it is satisfiable) and the answer "false" denotes a rejection of the problem instance (it is not satisfiable). One auxiliary function of the form $\mathrm{contract}(G, n_1, n_2)$ is used which gives the graph resulting from contracting the nodes $n_1, n_2$ in the graph $G$.

**Theorem 1.** *$PSAT_{\mathrm{po}}(\mathcal{A}_{14})$ is tractable.*

*Proof.* We begin by showing that algorithm PO solves $\mathrm{PSAT}_{\mathrm{po}}(\mathcal{C})$ correctly.

If the algorithm contracts two nodes $n_1, n_2$ in a graph $\Pi$, then $\Pi$ is satisfiable iff the contracted graph is satisfiable by Lemma 2 since either $n_1 \leq^* n_2, n_2 \leq^* n_1$ or $n_1 \underline{||} n_2, n_2 \leq^* n_1$ which both lead to $M(n_1) = M(n_2)$ in every model $M$ of $\Pi$. Thus, it is easy to see that if the algorithm rejects the graph $\Pi$ after $n$ iterations, the original instance cannot have a model.

Assume to the contrary that the algorithm accepts an instance. Consider the graph $\Pi$ after the last iteration and recall that the original graph is satisfiable iff $\Pi$ is satisfiable. The $(\leq)$-subgraph of $\Pi$ is acyclic and there exist no $n_1, n_2$ such that $n_1 \underline{||} n_2$ and $n_1 \leq^* n_2$. Let,

$$M = \langle f, \langle T, \Re \rangle \rangle = \langle \{\langle n_i, p_i \rangle | n_i \in \Pi\}, \langle \{p_i | n_i \in \Pi\}, \{\langle p_i, p_j \rangle | n_i \leq^* n_j\} \rangle \rangle$$

and note that $\langle T, \Re \rangle$ is a partial order. Arbitrarily choose $\langle x, R, y \rangle$ in $\Pi$ and assume without loss of generality that $x, y$ are distinct nodes. If $R$ is $\neq$ the relation holds since every node $n_i$ is mapped to a unique point $p_i$, if $R$ is $\underline{||}$ the relation holds since neither $x \leq^* y$ nor $y \leq^* x$ in $\Pi$. The last case when $R$ is $\leq$ trivially holds. Consequently $M$ is a model of $\Pi$ and the original problem instance is satisfiable. The algorithm trivially runs in polynomial time since every loop of the algorithm removes at least one node from the graph.

Finally, we note that $\mathcal{C}$ implements $\mathcal{A}_{14}$ by the following gadgets and the tractability of $\mathrm{PSAT}_{\mathrm{po}}(\mathcal{A}_{14})$ follows from Lemma 1.

$$\begin{tikzcd} x \arrow[r, bend left, "\leq"] \arrow[r, leftrightarrow, bend right, "\neq"'] & y \end{tikzcd} \Rightarrow x\{<\}y \qquad \begin{tikzcd} x \arrow[r, leftrightarrow, bend left, "\underline{\|}"] \arrow[r, leftrightarrow, bend right, "\neq"'] & y \end{tikzcd} \Rightarrow x\{\|\}y$$

$$\begin{tikzcd} x \arrow[r, bend left, "\leq"] & z \arrow[r, leftrightarrow, bend left, "\underline{\|}"] & y \end{tikzcd} \Rightarrow x\{<= \|\}y \qquad \begin{tikzcd} x \arrow[rr, leftrightarrow, bend left, "\neq"] \arrow[r, bend right, "\leq"'] & z \arrow[r, leftrightarrow, bend right, "\underline{\|}"'] & y \end{tikzcd} \Rightarrow x\{\| <\}y$$

$$\begin{tikzcd} x \arrow[r, bend left, "\leq"] & y \arrow[l, bend left, "\leq"] \end{tikzcd} \Rightarrow x\{=\}y$$

□

An alternative algorithm for $\mathrm{PSAT}_{\mathrm{po}}(\mathcal{A}_{14})$ is given in Anger *et al.* [2]

**Theorem 2.** *$PSAT_{\mathrm{po}}(\mathcal{A}_{10})$ is tractable.*

*Proof.* We begin by showing that Algorithm PO correctly solves the $\mathrm{PSAT}_{\mathrm{po}}(\mathcal{D})$ problem. First note that lines 7–8 of the algorithm is never executed since $\underline{\|} \notin \mathcal{D}$. If the algorithm contracts two nodes $n_1, n_2$ in a graph $\Pi$, then $\Pi$ is satisfiable iff the contracted graph is satsfiable by Lemma 2 since there exists a cycle $n_1 \leq^* n_2 \leq^* n_1$ which leads to $M(n_1) = M(n_2)$ in every model $M$ of $\Pi$. After having made this observation, it is trival to show that if the algorithm rejects a graph $\Pi$, then it does not have a model.

Assume to the contrary that the algorithm accepts the graph $\Pi$ and let $\Pi'$ denote the graph after the last iteration of the algorithm. We can construct a model for $\Pi'$ by induction over the number of nodes in $\Pi'$. If $\Pi'$ is empty then $M = \langle \emptyset, \langle \emptyset, \emptyset \rangle \rangle$ is a model of $\Pi'$. Assume we can construct a model for every graph with $k$ or fewer nodes. Let $\Pi'$ be an arbitrary graph containing $k+1$ nodes. Note that the $(\leq)$-subgraph of $\Pi'$ is acyclic and thus contains at least one minimal node $x$. By the induction hypothesis there exists a model $M' = \langle f, \langle T, \Re \rangle \rangle$ of $\Pi' - \{x\}$. Let,

$$M = \langle (f | n \longmapsto c), \langle T \cup \{c\}, \Re \cup \{\langle c, x \rangle | x \in T\} \rangle \rangle$$

where $c$ is a fresh point. Clearly $M$ is a model of $\Pi'$ and thus $\Pi$ is satisfiable.

To conclude the theorem, $\mathcal{D}$ implements $\mathcal{A}_{10}$ by the following gadgets and the tractability of $\mathrm{PSAT}_{\mathrm{po}}(\mathcal{A}_{10})$ follows by Lemma 1. Note that the gadgets for $\{>\}, \{>=\}$ have been omitted since they can easily be expressed with $\{<\}, \{<=\}$, respectively.

$$x \underset{\{<\|>\}}{\overset{\{<=\}}{\rightleftarrows}} y \quad \Rightarrow x\{<\}y \qquad x \underset{\{<=\}}{\overset{\{<=\}}{\rightleftarrows}} y \quad \Rightarrow x = y$$

$$x \underset{\{<\|>\}}{\overset{\{<=>\}}{\rightleftarrows}} y \quad \Rightarrow x\{<>\}y$$

□

Finally, $\mathrm{PSAT_{po}}(\mathcal{A}_9)$ is easily seen to be a tractable problem. Given an arbitrary instance, note that if two variables are related by the empty relation $\perp$, then the instance is not satisfiable. Otherwise, it has the trivial model which maps every node onto the same point.

### 3.2 The classification

The proof of the claim that there exists no other maximal tractable subclasses of $\mathrm{PSAT_{po}}(\mathcal{PA})$ than those presented in the previous subsection is given in Theorem 3. Before that, we need a number of auxiliary results, though.

We begin by noting that $\mathrm{PSAT_{po}}(\mathcal{PA})$ is in NP. This follows immediately since we can nondeterministically guess a model $M$ and verify it in polynomial time; it is sufficient to check the following: (1) $M$ is a partial order; and (2) each constraint $x\ R\ y$ holds under the interpretation.

Our hardness results rely on the fact that $\mathrm{PSAT_{po}}(\mathcal{A}_2)$ is NP-complete where $\mathcal{A}_2 = \{\{<>\}, \{\|\}\}$. We know that $\mathrm{PSAT_{po}}(\mathcal{A}_2)$ is in NP and we show hardness for NP by a polynomial-time reduction from the BETWEENNESS problem which is defined as follows:.

INSTANCE: A pair $\langle A, T\rangle$ consisting of a finite set $A$ and a collection $T$ of ordered triples $\langle a, b, c\rangle$ of distinct elements from $A$.
QUESTION: Is there a one-to-one mapping $f : A \to \{1, 2, \ldots, |A|\}$ such that for each $\langle a, b, c\rangle \in T$ we have either $f(a) < f(b) < f(c)$ or $f(c) < f(b) < f(a)$.

The BETWEENNESS problem is known to be NP-complete [7, page 279].

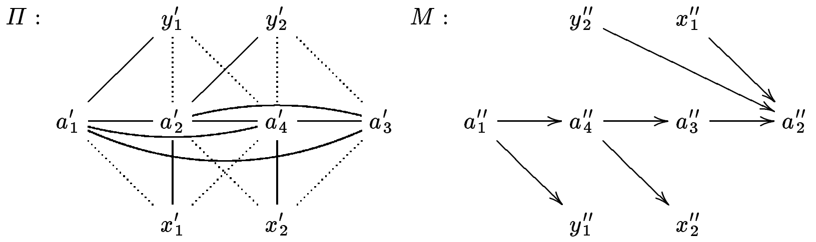
Given an arbitrary instance $\langle A, T\rangle$ of BETWEENNESS where $A = \{a_1, \ldots, a_k\}$, construct an instance $\Pi$ of $\mathrm{PSAT_{po}}(\mathcal{A}_2)$ as follows:

1. for each $a_i \in A$, introduce a fresh variable $a'_i$;
2. for each pair of distinct $a_i, a_j \in A$, add the constraint $a_i\{<>\}a_j$;
3. for each triple $t_m = \langle a_i, a_j, a_k\rangle \in T$, introduce two fresh variables $x'_m, y'_m$ and add the following constraints:

$$x'_m\{\|\}a'_i, x'_m\{\|\}a'_j, x'_m\{<>\}a'_k$$

$$y'_m\{<>\}a'_i, y'_m\{\|\}a'_j, y'_m\{\|\}a'_k$$

**Fig. 2.** The problem instance $\Pi$ and its model $M$. To clarify the picture, dotted arrows have been used to mark the relation $\{\|\}$ and filled arrows for the relation $\{<>\}$.




We show that $\Pi$ is satisfiable iff $\langle A, T\rangle$ has a solution. For the if-direction, assume $g : A \to \{1, \ldots, k\}$ is a solution to $\langle A, T\rangle$. Let $M = \langle f, \langle T, \Re\rangle\rangle$ where,

$$
\begin{aligned}
&f = \{\langle a'_i, a''_i\rangle | a'_i \in \Pi\} \\
&T = \{a''_i | a'_i \in \Pi\} \\
&\Re = \{\langle a''_i, a''_j\rangle | a_i, a_j \in A, g(a_i) < g(a_j)\} && (1)\\
&\quad \cup \{\langle x''_m, a''_k\rangle, \langle a''_i, y''_m\rangle | t_m = \langle a_i, a_j, a_k\rangle \in T, g(a_i) < g(a_j)\} && (2)\\
&\quad \cup \{\langle a''_k, x''_m\rangle, \langle y''_m, a''_i\rangle | t_m = \langle a_i, a_j, a_k\rangle \in T, g(a_i) > g(a_j)\} && (3)\\
&\quad \cup \{\langle x, y\rangle | \exists z : \langle x, z\rangle, \langle z, y\rangle \in \Re\} && (4)
\end{aligned}
$$

$M$ is a model of $\Pi$ since (1) satisfies all the constraints $a_i\{<>\}a_j$ introduced by step 2 above and (2), (3) satisfies the constraints introduced by step 3 above. (4) ensures that the transitive closure of $\Re$ is preserved.

To clarify the steps of the model construction above we provide a small example. The BETWEENNESS problem instance

$$\langle\{a_1, a_2, a_3, a_4\}, \{\langle a_1, a_4, a_2\rangle, \langle a_2, a_3, a_4\rangle\}\rangle$$

gives the point algebra problem instance $\Pi$ in figure 2 and has a solution $g = \{\langle a_1, 1\rangle, \langle a_2, 4\rangle, \langle a_3, 3\rangle, \langle a_4, 2\rangle\}$. From the solution of the BETWEENNESS problem we can construct a model $M$ of $\Pi$ depicted in the same figure. Transitive edges are omitted.

We continue by showing the only-if direction. Assume $\Pi$ has a model $M$. Note that the variables $a'_1, a'_2, \ldots$ in $\Pi$ corresponding to variables $a_1, a_2, \ldots$ in $A$ are totally ordered, *i.e.*, for arbitrary distinct $a'_i, a'_j$, either $M(a'_i) < M(a'_j)$ or $M(a'_j) < M(a'_i)$. Choose indices $m_1, \ldots, m_k$ such that $a'_{m_1} < \cdots < a'_{m_k}$. Define a solution $f : A \to \{1, \ldots, k\}$ of $\langle A, T\rangle$ as follows: $f(a_i) = m_i$. Clearly, this function is a one-to-one mapping from $A$ to $\{1, \ldots, k\}$

Arbitrarily choose a triple $\langle a_i, a_j, a_k\rangle \in T$. We show that $f(a_i) < f(a_j) < f(a_k)$ or $f(a_k) < f(a_j) < f(a_i)$. Assume to the contrary that $f(a_j) < f(a_i) < f(a_k)$. Then, the model $M$ implies that $M(a'_j) < M(a'_i) < M(a'_k)$ under the partial order of $M$. By the construction of $\Pi$ we know that there exists a variable

$y'_m$ such that $M(y'_m)\ \{<>\}\ M(a'_i)$, $M(y'_m)\ \{\|\}\ M(a'_j)$ and $M(y'_m)\ \{\|\}\ M(a'_k)$. Assume first that $M(y'_m)\ \{<\}\ M(a'_i)$, then $M(y'_m)\ \{<\}\ M(a'_k)$ which contradicts the fact that $M(y'_m)\ \{\|\}\ M(a'_k)$. Similary, if $M(y'_m)\ \{>\}\ M(a'_i)$ then $M(a'_j)\ \{<\}\ M(y'_m)$ and we have a contradiction since $M(y'_m)\ \{\|\}\ M(a'_j)$. The remaining three cases can be proven analogously.

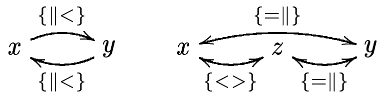
Thus, we have proven the following lemma.

**Lemma 3.** *$PSAT_{\mathrm{po}}(\mathcal{A}_2)$ is NP-complete.*

We also need an auxiliary result that proves NP-completeness of all classes containing $\{<>\}$ and a relation not present in $\mathcal{A}_{10}$.

**Lemma 4.** *If $C$ is a set of relations containing $\{<>\}$ and one of $\{\| <\}, \{= \|\}$ or $\{<= \|\}$, then $PSAT_{\mathrm{po}}(C)$ is NP-complete.*

*Proof.* Assume $\{\| <\}$ or $\{= \|\}$ is in $C$. Then, the following gadgets show that $C$ implements $\{\|\}$ and NP-completeness follows by Lemma 1 and Lemma 3.




If $C$ contains $\{<= \|\}$, then it is easy to show that $C$ implements $\{= \|\}$ and NP-completeness follows from the previous case. $\square$

We can now give the complete classification of the point algebra for partially-ordered time.

**Theorem 3.** *The classes $\mathcal{A}_{14}$, $\mathcal{A}_{10}$ and $\mathcal{A}_9$ are the only maximal tractable subclasses of $PSAT_{\mathrm{po}}(\mathcal{PA})$.*

*Proof.* Assume there exists a maximal tractable subclass $C$ of $\mathcal{PA}$ which is not $\mathcal{A}_{14}$, $\mathcal{A}_{10}$ nor $\mathcal{A}_9$. Then $C$ must contain (not neccessarily distinct) elements which are not present in either one of those classes. That is,

$$\exists x, y, z \in C : x \in \mathcal{A}_{14}^{-1}, y \in \mathcal{A}_{10}^{-1}, z \in \mathcal{A}_9^{-1}$$

where

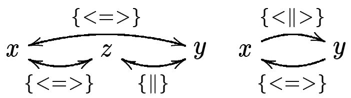
$$\mathcal{A}_{14}^{-1} = \{\{<>\}, \{<=>\}\}$$
$$\mathcal{A}_{10}^{-1} \equiv \{\{\|\}, \{\| <\}, \{= \|\}, \{<= \|\}\}$$
$$\mathcal{A}_9^{-1} \equiv \{\{<\}, \{<>\}, \{\|\}, \{\| <\}, \{< \| >\}\}$$

Note that the relations $\{>\}, \{> \|\}$ and $\{>= \|\}$ have been omitted since they can trivially be expressed with $\{<\}, \{\| <\}$ and $\{<= \|\}$, respectively.

Assume that $\{<>\} \in C$. By Lemma 3 and Lemma 4, $C$ is NP-complete, since one element in $\mathcal{A}_{10}^{-1}$ must be in $C$. Contradiction, and it follows that $\{<=>\} \in C$.

We know that $C$ contains $\{<=>\}$, one element from $\{\{<\}, \{\|\}, \{\| <\}, \{< \| >\}\}$ and one element from $\{\{\|\}, \{\| <\}, \{= \|\}, \{<= \|\}\}$. First we consider two cases that eliminates the relations $\{\|\}, \{< \| >\}$ and $\{\| <\}$.

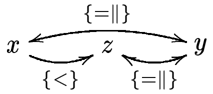
(1) If $\{\|\} \in C$ or $\{< \| >\} \in C$ then the following gadgets implement $\{<>\}$ which contradicts our previously made assumptions.




(2) If $\{\| <\} \in C$ then it is trivial to implement $\{\|\}$ which leads to a contradiction by (1).

Now we know that $\{<=>\}, \{<\} \in C$ and that either $\{= \|\}$ or $\{<= \|\}$ is in $C$. We show that the last two cases also leads to a contradiction which prove the theorem.

(3) If $\{<\} \in C$ and $\{= \|\} \in C$, then the following gadget implements $\{\|\}$ which leads to a contradiction by (1).




(4) If $\{<\} \in C$ and $\{<= \|\} \in C$, then it is trivial to implement $\{= \|\}$ which lead to a contradiction by (3).

□

## 4 Branching Time

When examining the case of branching time, we begin by studying a small set $\mathcal{E}$ containing the three relations $\leq, \|, \neq$. We will show that $\mathrm{PSAT}_{\mathrm{br}}(\mathcal{E})$ is tractable and that $\mathcal{E}$ implements the relations in the set $\mathcal{A}_{13}$. Consequently, $\mathrm{PSAT}_{\mathrm{br}}(\mathcal{A}_{13})$ is tractable. It should be noted that $\mathcal{A}_{13}$ is a nontrivial subclass which, for instance, contains all basic relations.

A node $n$ is said to be *superminimal* in a problem instance $\Pi$ iff $n$ is a $(\leq)$-minimal node of $\Pi$ and there exists no $x$ such that $n\|x$. A *semipath* between two nodes $n_1, n_2$ in a graph $\Pi$ is a set of nodes $a_1, \ldots, a_k, b_1, \ldots, b_k$ such that:

$$n_1 \leq^* a_1 \geq^* b_1 \leq^* \ldots \leq^* a_k \geq^* b_k \leq^* n_2$$

It should be noted that iff two nodes $n_1, n_2$ are part of the same $(\leq)$-component in a graph $\Pi$ then there exists a semipath between $n_1, n_2$.

Before we can prove the main result of this section, we need several lemmata.

**Lemma 5.** *If $a \leq^* b$ and $b(<=>)c$, then $a(<=>)c$.*

*Proof.* Immediate consequence of the composition table for branching time □

In the point algebra of branching time we have an important propery which helps us prove correctness of the algorithm BR. Only superminimal nodes in a problem instance $\Pi$ may be mapped to a minimal point in any model of $\Pi$, which the following lemma proves.

**Lemma 6.** *Assume that $\Pi$ is a problem instance containing nodes $n_1, n_2$ such that $n_1 \| n_2$ and there exists a semipath between $n_1, n_2$, the image of $n_1$ cannot be a minimal node in any model of $\Pi$.*

*Proof.* Assume $M$ is a model of $\Pi$ such that $M(n_1)$ is a minimal node. Since there exists a semipath between $n_1, n_2$ there exists $a_1, \ldots, a_k, b_1, \ldots, b_k$ such that:

$$n_1 \leq^* a_1 \geq^* b_1 \leq^* \ldots \leq^* a_k \geq^* b_k \leq^* n_2$$

Lemma 5 gives $n_1 (<=>) b_1$, $M(n_1)$ minimal gives $n_1 \leq b_1$. Note that $n_1 \leq b_1 \leq^* a_2$ and $b_2 \leq^* a_2$, Lemma 5 gives $n_1 (<=>) b_2$ thus $n_1 \leq b_2$. Repeat for each $b$ in $b_2, \cdots, b_k$ and we have $n_1 \leq b_k \leq^* n_2$. Hence $n_1 \leq^* n_2$ and $n_1 \| n_2$, contradiction. Thus there exists no model $M$ such that $M(n_1)$ is a minimal node. □

A model $M = \langle f, \langle T, \Re \rangle \rangle$ of $\Pi$ is said to be *non-redundant* iff $f$ is surjective, ie. every point in $M$ is the image of some node in $\Pi$. Let $\Pi$ be a problem instance of $\mathrm{PSAT_p}(\Re)$. If $\Pi$ is satisfiable then there exists a non-redundant model of $\Pi$. Assume $\Pi$ is satisfiable, then there exists a model $M = \langle f, \langle T, \Re \rangle \rangle$. Let $T'$ be the points in $T$ that are the image of some node in $\Pi$ and let

$$M' = \langle f, \langle T', \Re \cap (T' \times T') \rangle \rangle$$

Clearly $M'$ is a model of $\Pi$ and $M'$ is non-redundant.

**Lemma 7.** *Let $\Pi$ be a problem instance of $PSAT_{\mathrm{br}}(\mathcal{E})$ such that $\Pi$ is $(\leq)$-acyclic and $M = \langle f, \langle T, \leq \rangle \rangle$ is a non-redundant model of $\Pi$. Then there exists at least one minimal point $p$ in $\langle T, \leq \rangle$ which is the image of a $(\leq)$-minimal node in $\Pi$.*

*Proof.* Since $\langle T, \leq \rangle$ is a finite partial order, it contains a minimal point $p$. Let $N = n_1, \cdots, n_k$ be the nodes in $\Pi$ that are mapped to $p$ by $f$. We know that $M$ is nonredundant so $N \neq \emptyset$. Assume now that no member of $N$ is $(\leq)$-minimal. $\Pi$ is acylic so there exists a $(\leq)$-minimal node $n'$ such that $n' \leq^* n_1 \in N$. The $(\leq)$-minimality of $n'$ implies that $n' \notin N$ and $f(n') \neq p$. This leads immediatly to a contradiction since $f(n') < f(n_1) = p$ and $p$ is minimal in $\langle T, \leq \rangle$. Consequently, there exists a $(\leq)$-minimal node in $N$ that is mapped to $p$. □

Now, we can prove that algorithm BR correctly solves $\mathrm{PSAT_{br}}(\mathcal{E})$.

**Lemma 8.** *The algorithm in Fig. 3 correctly solve the $PSAT_{\mathrm{br}}(\mathcal{E})$ problem.*

*Proof.* We prove correctness of the algorithm by induction over the number of nodes in the instance. The algorithm accepts all instances $\Pi$ containing zero nodes since $M = \langle \{\emptyset\}, \langle \emptyset, \{\emptyset\} \rangle \rangle$ is a model of every such instance.

Assume the algorithm correctly solves the problem for every graph containing $k$ nodes. Let $\Pi$ be an arbitrary graph of $k+1$ nodes. We consider four cases:

1. (lines 5-7) There exists two nodes $n_1, n_2$ such that $n_1 \leq^* n_2, n_2 \leq^* n_1$. If $n_1 \neq n_2$ in $\Pi$, then there can be no model of $\Pi$ and the algorithm rejects. Otherwise, by Lemma 2 $\Pi$ is satisfiable iff contract$(\Pi, n_1, n_2)$ is satisfiable and the algorithm is correct by the induction hypothesis.
2. (lines 10-11) There exists at least two $(\leq)$-components $C_1, \cdots, C_n$ of $\Pi$. If the algorithm rejects some $C_i$ then $C_i$ cannot have a model by the induction hypothesis. Since $C_i$ is a subgraph of $\Pi$, there exist no model of $\Pi$.
   We show that if $C_i$ is satisfiable for every $i$ then $\Pi$ is satisfiable. For each $i \leq n$, let $M_i = \langle f_i, \langle T_i, \Re_i \rangle \rangle$ denote a model for $C_i$ and let
   $$M = \langle f_1 \cup f_2 \cup \cdots \cup f_n, \langle T_1 \cup T_2 \cup \cdots \cup T_n, \Re_1 \cup \Re_2 \cup \cdots \Re_n \rangle \rangle$$
   Clearly $M$ in an interpretation of $\Pi$ since every $M_i$ is an interpretation of $C_i$. The only constraint between two distinct nodes $n_i \in C_i, n_j \in C_j$ can be $\neq$ or $\|$ which trivially holds so $M$ is a model of $\Pi$.
3. (lines 12-13) There exists only one $(\leq)$-component of $\Pi$ which has a superminimal node $n$. If the algorithm rejects then $\Pi - \{n\}$ is unsatisfiable by the induction hypothesis and $\Pi$ cannot have a model.
   If $\Pi - \{n\}$ is satisfiable, then $\Pi$ is satisfiable: let $M' = \langle f, \langle T, \Re \rangle \rangle$ denote a model of $\Pi - \{n\}$ and let
   $$M = \langle (f | n \longmapsto c), \langle T \cup \{c\}, \Re \cup \{\langle c, x \rangle | x \in T\} \rangle \rangle$$
   where $c$ is a fresh point. Clearly $M$ is an interpretation of $\Pi$ since the only point $c$ added to $M'$ is a minimal point. Assume $\langle n, R, v \rangle \in \Pi$. If $R$ is $\neq$, then the relation holds since $f(n)$ is a fresh point. Otherwise $R$ is $\leq$ since the superminimality of $n$ prohibits $R$ from being either $\|$ or $\geq$. Hence, the relation holds since $f(n)$ is a minimal point and $M$ is a model of $\Pi$.
4. (line 15) There exists only one $(\leq)$-component of $\Pi$ and $\Pi$ contains no superminimal node. Assume $\Pi$ is satisfiable. Then, there exists a non-redundant model $M$ of $\Pi$. By Lemma 7, $M$ has a minimal point $p$ which is the image of some $(\leq)$-minimal node $n$. Since $n$ is a minimal node but not a superminimal node there exists an $x$ such that $n \| x$. Since $\Pi$ only has one component there exists a $(\leq)$-semipath between $n$ and $x$. By Lemma 6 the image $p$ of $n$ cannot be a minimal node which is a contradiction. Hence, there cannot exist a model of $\Pi$.

□

**Lemma 9.** *$PSAT_{\mathrm{br}}(\mathcal{E})$ is tractable.*

*Proof.* The correctness of algorithm BR for the $\mathrm{PSAT}_{\mathrm{br}}(\mathcal{E})$ problem follows from Lemma 8. To show that BR runs in polynomial time, let $p(n)$ be a polynomial that is an upper bound for the time complexity of which steps 5 and 9 can be performed. The time complexity of the algorithm is bounded from above by the solution to the following recursive equation:

```
1   Algorithm BR
2   Input  Graph Π
3   if Π is empty  then
4     return true
5   elsif ∃n1, n2 : n1 ≤* n2, n2 ≤* n1  then
6     if n1 ≠ n2  then return false
7     else return BR(contract(Π,n1,n2))
8   else
9     Identify the (≤)-components C1, ···, Ck of Π.
10    if k > 1  then
11      return BR(C1) ∧ BR(C2) ∧ ··· ∧ BR(Ck)
12    elsif ∃n : superminimal(n)  then
13      return BR(Π − {n})
14    else
15      return false
16    end if
17  end if
```

**Fig. 3.** The algorithm for solving $\mathrm{PSAT}_{\mathrm{br}}(\mathcal{E})$

$$f(1) = 1$$
$$f(n) = \max \begin{cases} p(n) + f(n-1) \\ p(n) + \sum_{i=1}^{k} f(c_i) : k \geq 2, c_i > 0, \sum_{i=1}^{k} c_i = n \end{cases}$$

The case $f(n) = p(n) + f(n-1)$ occurs when the algorithm recurse at line 13 and the case $f(n) = \sum f(c_i)$ occurs when the algorithm recurse at line 11 (where $c_i$ denotes the size of component $C_i$). It can be shown that $f(n) \leq n^2 p(n)$ by a straightforward induction over $n$. $\square$

**Theorem 4.** *$PSAT_{\mathrm{br}}(\mathcal{A}_{13})$ is tractable.*

*Proof.* The following gadgets prove that $\mathcal{E}$ implements $\mathcal{A}_{13}$, by the previous lemma $\mathrm{PSAT}_{\mathrm{br}}(\mathcal{E})$ is tractable. Hence, $\mathrm{PSAT}_{\mathrm{br}}(\mathcal{A}_{13})$ is tractable.

$$\begin{tikzcd} x \arrow[r, bend left, "\leq"] \arrow[r, bend right, leftrightarrow, "\neq"'] & y \end{tikzcd} \Rightarrow x\{<\}y \qquad \begin{tikzcd} x \arrow[rr, bend left, leftrightarrow, "\neq"] \arrow[r, bend right, "\leq"'] & z & y \arrow[l, bend left, "\leq"] \end{tikzcd} \Rightarrow x\{<>\}y$$

$$\begin{tikzcd} x \arrow[r, bend left, "\leq"] & z & y \arrow[l, bend right, "\leq"'] \end{tikzcd} \Rightarrow x\{<=>\}y \qquad \begin{tikzcd} x \arrow[r, bend left, "\leq"] & y \arrow[l, bend left, "\leq"] \end{tikzcd} \Rightarrow x\{=\}y$$

$$\begin{tikzcd} x \arrow[r, bend left, "\|"] & z \arrow[r, bend left, "\leq"] & y \end{tikzcd} \Rightarrow x\{\| <\}y$$

$\square$

## 5 Concluding Remarks

We have identified all tractable subclasses of the point algebra for partially-ordered time and studied one large tractable subclass of the point algebra for branching time. In the case of branching time, there is one major question left open by this paper: is $\mathcal{A}_{13}$ a maximal tractable subclass of $\mathrm{PSAT}_{\mathrm{br}}(\mathcal{PA})$? Or is $\mathrm{PSAT}_{\mathrm{br}}(\mathcal{PA})$ itself tractable? In the case of partially-ordered time, several minor questions are still to be answered. One quite interesting question is whether path-consistency is enough to decide satisfiability of the tractable classes.

## References


1. F. Anger. On Lamport's interprocessor communication model. *ACM Transactions on Programming Languages Systems*, 11(3):404–417, 1989.
2. F. Anger, D. Mitra, and R. Rodriguez. Temporal constraint networks in nonliear time. Technical report, ECAI Workshop on Temporal and Spatial Reasoning, 1998.
3. M. Broxvall. Computational complexity of point algebras for nonlinear time. Master Thesis Report LiTH-IDA-Ex-99/48, Department of Computer Science, Linköping, Sweden, June 1999.
4. E. Clarke, E. Emerson, and A. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages Systems*, 8(2):244–263, 1986.
5. T. Dean and M. Boddy. Reasoning about partially ordered events. *Artificial Intelligence*, 36:375–399, 1988.
6. A. E. Emerson and J. Y. Halpern. "Sometimes" and "not never" revisited: On branching versus linear temporal logic. *Journal of the ACM*, 33(1):151–178, Jan. 1986.
7. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, New York, 1979.
8. P. Jonsson and T. Drakengren. A complete classification of tractability in RCC-5. *Journal of Artificial Intelligence Research*, 6:211–221, 1997.
9. L. Lamport. The mutual exclusion problem: Part I—a theory of interprocess communication. *Journal of the ACM*, 33(2):313–326, 1986.
10. D. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.
11. B. Nebel and H.-J. Bürckert. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *Journal of the ACM*, 42(1):43–66, 1995.