Twenty-one Large Tractable Subclasses of Allen's Algebra

Thomas Drakengren and Peter Jonsson

ABSTRACT

This paper continues Nebel and Bürckert's investigation of Allen's interval algebra by presenting nine more maximal tractable subclasses of the algebra (provided that $P \neq NP$), in addition to their previously reported *ORD-Horn* subclass. Furthermore, twelve tractable subclasses are identified, whose maximality is not decided. Four of them can express the notion of *sequentiality* between intervals, which is not possible in the ORD-Horn algebra. All of the algebras are considerably larger than the ORD-Horn subclass. The satisfiability algorithm, which is common for all the algebras, is shown to be linear. Furthermore, the path consistency algorithm is shown to decide satisfiability of interval networks using any of the algebras.

1 Introduction

For specifying qualitative temporal information about relations between intervals, Allen's interval algebra [Allen, 1983] is often considered a convenient tool. However, due to its expressiveness (the satisfiability problem is NP-complete [Vilain *et al.*, 1989]), it is unlikely that there will be a polynomial-time algorithm for reasoning about the full algebra. Trying to overcome this, several tractable fragments of the algebra have been identified (e.g. [Nebel and Bürckert, 1995; van Beek, 1989; Golumbic and Shamir, 1993]), of which the largest known is Nebel and Bürckert's *ORD-Horn* algebra [Nebel and Bürckert, 1995]. Furthermore, this algebra has been proved to be the *unique maximal* algebra containing all the basic relations, comprising approximately 10 percent of the full algebra.

None of these algebras, however, is capable of expressing the notion of *sequentiality*, which is that of specifying that some intervals have to occur in sequence in time, without any overlap. This is required e.g. in some cases of reasoning about action [Sandewall, 1994]. The maximality result of the ORD-Horn algebra then implies that the requirement that an algebra should contain all the basic relations has to be sacrificed. Golumbic and Shamir [1993] come close to expressing sequentiality, but require that *any* two intervals are related, except for in an almost trivial four-element subset, which is not even an algebra. Four of our algebras strictly extend this subset.

In this paper, we exploit a simple graph algorithm, similar to that of van Beek [1992], and show that we can construct 21 large algebras for which this algorithm solves satisfiability in linear time, and furthermore, that four of these can express sequentiality, and nine of them are *maximal* tractable algebras (assuming $P \neq NP$, which we take for true in the rest of the paper). It should be noted that these algebras are of a size considerably larger than the ORD-Horn algebra: 20 contains 2178 elements, and one 4097 elements, which is half of Allen's algebra. However, this largest algebra has almost no expressiveness, showing that the size of an algebra need not have anything to do with its usefulness (as has often been argued in context with the ORD-Horn algebra).

The structure of the paper follows. In Section 2 and Section 3 we present the necessary background material about Allen's interval algebra, and some facts about the ORD-Horn algebra. Then, in Section 4, the concepts of "acyclic" and "DAG-satisfying" relations are introduced, after which the main results of the new tractable algebras are presented in Section 5. We show how the algorithm works in Section 6, and finally in Section 7, we show

Basic relation		Example	Endpoints
X before Y	$\prec$	xxx	$X^+ < Y^-$
Y after X	$\succ$	yyy	
X meets Y	m	xxxx	$X^+ = Y^-$
Y met-by X	$\mathsf{m}^{\smile}$	yyyy	
X overlaps Y	o	xxxx	$X^- < Y^- < X^+$,
Y overlapped-by X	$\mathsf{o}^{\smile}$	yyyy	$X^+ < Y^+$
X during Y	d	xxx	$X^- > Y^-$,
Y includes X	$\mathsf{d}^{\smile}$	yyyyyyy	$X^+ < Y^+$
X starts Y	s	xxx	$X^- = Y^-$,
Y started-by X	$\mathsf{s}^{\smile}$	yyyyyyy	$X^+ < Y^+$
X finishes Y	f	xxx	$X^+ = Y^+$,
Y finished-by X	$\mathsf{f}^{\smile}$	yyyyyyy	$X^- > Y^-$
X equals Y	$\equiv$	xxxx	$X^- = Y^-$,
		yyyy	$X^+ = Y^+$

Table 1: The thirteen basic relations. The endpoint relations $X^- < X^+$ and $Y^- < Y^+$ that are valid for all relations have been omitted.

that the *path consistency algorithm* is sufficient for deciding consistency for any of these algebras. A discussion concludes the paper.

This paper is an extended and modified version of an earlier paper [Drakengren and Jonsson, 1996]. The main additions are Section 6 and Section 7, but we have also improved the presentation of the algebras considerably.

2 Allen's Interval Algebra

Allen's interval algebra [Allen, 1983] is based on the notion of *relations between pairs of intervals*. An interval X is represented as an ordered pair $\langle X^-, X^+\rangle$ of real numbers with $X^- < X^+$, denoting the left and right endpoints of the interval, respectively, and relations between intervals are composed as disjunctions of *basic interval relations*, which are those in Table 1. Such disjunctions are represented as *sets* of basic relations, but using a notation such that, for example, the disjunction of the basic interval relations $\prec$, m and $\mathsf{f}^{\smile}$ is written $(\prec\ \mathsf{m}\ \mathsf{f}^{\smile})$. Thus, we have that $(\prec\ \mathsf{f}^{\smile}) \subseteq (\prec\ \mathsf{m}\ \mathsf{f}^{\smile})$.

The algebra is provided with the operations of *converse, intersection and* *composition* on interval relations:

- The converse operation takes an interval relation i to its converse $i^{\smile}$, obtained by inverting each basic relation in i, by exchanging X and Y in the endpoint relations of Table 1.

- The intersection operation takes two interval relations R_1 and R_2 to their intersection $R_1 \cap R_2$, by taking the basic relations that are contained in both R_1 and R_2.

- The composition operation takes two interval relations R_1 and R_2 to their composition $R_1 \circ R_2$, which is the Allen relation R_3 such that IR_3J iff $\exists K.IR_1K \wedge KR_2J$.

By the fact that there are thirteen basic relations, we get $2^{13} = 8192$ possible relations between intervals in the full algebra. We denote the set of all interval relations by $\mathcal{A}$. Subclasses of the full algebra are obtained by considering subsets of $\mathcal{A}$.

Although there are several computational problems associated with Allen's interval algebra, this paper focuses on the problem of *satisfiability* of a set of interval variables with relations between them, i.e. deciding whether there exists an assignment of intervals on the real line for the interval variables, such that all of the relations between the intervals hold. We define this as follows.

Definition 2.1 ($ISAT(S)$) Let $S \subseteq \mathcal{A}$ be a set of interval relations. An instance of $ISAT(S)$ is a labelled directed graph $G = \langle V, E\rangle$, where the nodes in V are interval variables and E is a subset of $V \times S \times V$. A labelled edge $\langle u, r, v\rangle \in E$ means that u and v are related by the relation r.

A function M taking an interval variable v to its interval representation $M(v) = \langle x^-, x^+\rangle$ with $x^- < x^+$, is said to be an *interpretation* of G.

An instance $G = \langle V, E\rangle$ is said to be *satisfiable* iff there exists an interpretation M such that for each $\langle u, r, v\rangle \in E$, $M(u)rM(v)$ holds, i.e., the endpoint relations required by r (see Table 1) are satisfied by the assignments of u and v. Then M is said to be a *model* of G.

We refer to the *size* of an instance G as $|V| + |E|$, as usual. $\square$

For $\mathcal{A}$, we have the following result.

Proposition 2.2 $ISAT(\mathcal{A})$ is NP-complete.
Proof: See Vilain et al [1989]. $\square$

The following auxiliary concept shall be needed.

Definition 2.3 ("satisfied as") Let $\mathcal{I} = \langle V, E \rangle$ be an instance of the satisfiability problem, M a model for $\mathcal{I}$, $\langle I_1, r, I_2 \rangle \in E$, and $r' \subseteq r$. Then r is said to be *satisfied as r' in M* iff $I_1 r' I_2$ is satisfied in M. □

Example 2.4 Let I_1, I_2 be interval variables related by $I_1(\prec \succ)I_2$, and M a model where I_1 is interpreted as $[1, 2]$ and I_2 as $[3, 4]$. Then in M, $(\prec \succ)$ is satisfied as $(\prec)$, but also as $(\prec \succ)$. □

3 The ORD-Horn Subclass

Nebel and Bürckert [1995] identify a subclass of the interval algebra, having the property that it is a maximal subclass containing all the basic interval relations, for which satisfiability can be solved using a polynomial-time algorithm, and is in fact the *unique* such maximal class[1]. This algebra, the *ORD-Horn* algebra, contains 868 relations, and thus covers slightly more than 10 percent of $\mathcal{A}$.

One of the main tools for analysing the ORD-Horn subclass is a *closure operation* on subclasses of the algebra, which preserves tractability.

Definition 3.1 (Closure) Let $S \subseteq \mathcal{A}$. Then we denote by $\overline{S}$ the *closure* of S under converse, intersection and composition, i.e. the least subalgebra containing S closed under the three operations. □

The key for extrapolating intractability results is the following.

Proposition 3.2 Let $S \subseteq \mathcal{A}$. Then $ISAT(S)$ is polynomial iff $ISAT(\overline{S})$ is, and $ISAT(S)$ is NP-complete iff $ISAT(\overline{S})$ is.
Proof: See Nebel and Bürckert [1995]. □

4 Acyclic and DAG-satisfying Relations

This section introduces some auxiliary notions and results needed for defining the new algebras, and proving their properties. In the rest of this paper, we let G denote an instance of $ISAT(X)$ instance for some $X \subseteq \mathcal{A}$.

[1] The uniqueness is proved under the assumption that the subclass must contain the empty relation ( ) and the full relation $(\equiv \prec \succ \mathsf{d}\, \mathsf{d}^{\smile}\, \mathsf{o}\, \mathsf{o}^{\smile}\, \mathsf{m}\, \mathsf{m}^{\smile}\, \mathsf{s}\, \mathsf{s}^{\smile}\, \mathsf{f}\, \mathsf{f}^{\smile})$.

Definition 4.1 (Acyclic relation) A relation r is said to be an *acyclic relation* iff for every G and every cycle C in G such that every arc in C is labelled with r, C is never satisfiable. $\square$

Example 4.2 $\prec$ is an acyclic relation, and so is $(\prec\ \mathsf{m})$. $\square$

Corollary 4.3 Let r be an acyclic relation. Then every relation $r' \subseteq r$ is acyclic.
Proof: Since taking subsets of r constrains satisfiability further, the result follows. $\square$

Corollary 4.4 Let r be an acyclic relation, and A such that $A \subseteq \{r' | r' \subseteq r\}$. Then, for every G and every cycle C in G such that every arc in C is labelled by some relation in A, C is unsatisfiable.
Proof: Same argument as in Corollary 4.3. $\square$

Definition 4.5 (Maximal acyclic relation) An acyclic relation r for which there is no acyclic relation $r' \supset r$, is said to be a *maximal acyclic relation.* $\square$

In Proposition 4.8, we shall list all possible maximal acyclic relations.

Proposition 4.6 Let r be an acyclic relation, and A, A' sets such that $A \subseteq \{r' | r' \subseteq r\}$, and $A' = \{a \cup (\equiv) | a \in A\} \cup \{(\equiv)\}$. Then, every cycle C labelled by relations in $A \cup A'$ is satisfiable iff it contains only relations from A', and, furthermore, in every model of C all relations in the cycle have to be satisfied as $\equiv$.
Proof:
$\Rightarrow$) Suppose that a cycle C is satisfiable, and that it contains some relation from A. Induction on the number n of arcs in the cycle. For $n = 1$, we get a contradiction by the assumption. So, suppose for the induction that C contains $n + 1$ arcs. Let M be a model for the relations in C. It cannot hold that every relation in C is satisfied in M as some relation in A, by Corollary 4.4. Thus, some relation r' in C has to be satisfied as $\equiv$. But then we can collapse the two interval variables connected by r' to one interval variable, and we have a cycle of size n containing a relation from A. This contradicts the induction hyphothesis.
$\Leftarrow$) Suppose that a cycle C contains only relations in A'. Then C can be satisfied by choosing $\equiv$ on every arc, thus forcing the satisfying intervals to be identical. $\square$

An example is in order.

Example 4.7 Let $r = (\prec \ \ \mathsf{m})$ (which is easily verified to be acyclic), $A = \{(\prec \ \ \mathsf{m}), (\prec), (\ )\}$, and thus $A' = \{(\equiv \prec \ \ \mathsf{m}), (\equiv \prec), (\equiv)\}$. Now take a cycle $I_1(\prec \ \mathsf{m})I_2(\equiv \prec)I_3(\equiv \prec \ \mathsf{m})I_1$. It is clear that this cycle is unsatisfiable, since it contains one relation which is not in A'. The cycle $I_1(\equiv \prec \ \mathsf{m})I_2(\equiv)I_3(\equiv \prec)I_1$ is clearly satisfiable taking $\equiv$ on each arc, but is not satisfiable in any other way, by the result. □

Next, we find all possible acyclic relations. The second part of the result, that there are no other maximal acyclic relations, is not used in any of the later results of the paper. However, it is important to know that there is no need to search for more maximal acyclic relations in future research.

Proposition 4.8 The only maximal acyclic relations in $\mathcal{A}$ are

$$(\prec \ \mathsf{d}^{\smile}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}\ \mathsf{f}^{\smile}), \quad (\prec \ \mathsf{d}^{\smile}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}^{\smile}\ \mathsf{f}^{\smile}),$$
$$(\prec \ \mathsf{d}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}\ \mathsf{f}), \quad (\prec \ \mathsf{d}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}\ \mathsf{f}^{\smile}),$$

and their respective converses.

Proof: Obviously, a maximal acyclic relation cannot contain both a basic relation and its converse, and thus cannot contain $\equiv$. One consequence of this is that a maximal acyclic relation cannot contain more than six basic relations. So, if the above relations are shown to be acyclic, then they are also maximal.

Now, consider Table 2, which extracts from Table 1 how the basic relations (except for $\equiv$) relate the ending points of intervals. The table is to be read as follows. Suppose that the intervals i_1 and i_2 are related by some basic relation b, i.e. $i_1(b)i_2$, and consider the l row entry for b.

- If it is $+$, then the starting point of i_2 must be strictly *after* the starting point of i_1
- If it is $-$ then the starting point of i_2 must be strictly *before* the starting point of i_1
- If it is $=$, then the starting points of i_1 and i_2 have to coincide.

Similarly, the r row states the same information for the ending points.

Now consider the l row. If we choose a relation r' to contain exactly the basic relations which have a $+$ there, we know that r' will be an acyclic relation, because if in a cycle, the left ending points of the intervals have to increase at every arc, it cannot be satisfied. In addition to those basic relations in r', we can include in r' one basic relation b' which has an $=$ in the l row, yielding the relation r'', since then, a cycle labelled by r'' on every arc has to be satisfied as b' on every arc (otherwise, we would get a contradiction, by strictly increasing starting point values). But since neither of s and $\mathsf{s}^{\smile}$ has an $=$ in their r row, this is impossible. This gives us two choices of acyclic relations, which are the two first ones listed.

Symmetrically, by inspecting the r row, we see that we get the next two relations listed. Finally, by taking the $-$ entries instead of the $+$ entries, we get the converse relations of the listed ones.

It remains to prove that these are the only maximal acyclic relations. So, suppose that some acyclic relation e is not a subset of (or equal to) any of the relations in the statement of the proposition. First, note that e cannot be a basic relation, since every basic acyclic relation is included in some of the listed relations. Thus, e has to contain at least two distinct basic relations b_1 and b_2. Without loss of generality (using Corollary 4.3), we have that $e = (b_1\ b_2)$.

By the choice of the listed relations, b_1 and b_2 must have opposite signs either in their l or r rows (or both). Suppose that b_1 and b_2 do not have opposite signs in their l row, i.e. that either they have the same sign, or at least one of them has an $=$. If both of them have an $=$, they have to be s and $\mathsf{s}^{\smile}$, which is impossible. If they have the same sign, which is not $=$, then they are included in one of the listed relations, by definition. If at least one of them, say b_1, has $=$ there, i.e. b_1 is either s or $\mathsf{s}^{\smile}$, we see that for any basic acyclic relation c, c and b_1 occur together in some of the listed relations (or their converses), and in particular, this holds when c is b_2. Thus b_1 and b_2 have to have opposite signs in the l row. Symmetrically, b_1 and b_2 must have opposite signs also in the r row.

Now, the only remaining choice of b_1 and b_2, for which the signs of the l and r rows do not coincide, is for the basic intervals d and $\mathsf{d}^{\smile}$. But trivially, these cannot together be part of any acyclic relation, and thus b_1 and b_2 have to be chosen such without loss of generality, b_1 has $+$ in both its l and r rows, and similarly for b_2, $-$ in both its l and r rows. Obviously, also every choice when b_1 and b_2 are converses is impossible.

This leaves us with six relations to check: $(\succ\ \mathsf{m})$, $(\mathsf{o}^{\smile}\ \mathsf{m})$ and $(\prec\ \mathsf{o}^{\smile})$ and their converses, and it is enough to check the first three due to symmetry. Now, it is easy to construct satisfiable cycles using relations containing either of these relations. $\square$

	m	m$^\smile$	$\prec$	$\succ$	o	o$^\smile$
l	$+$	$-$	$+$	$-$	$+$	$-$
r	$+$	$-$	$+$	$-$	$+$	$-$

	d	d$^\smile$	f	f$^\smile$	s	s$^\smile$
l	$-$	$+$	$-$	$+$	$=$	$=$
r	$+$	$-$	$=$	$=$	$+$	$-$

Table 2: The effect of relations on interval endpoints.

Definition 4.9 (DAG-satisfying relation) A basic relation b is said to be *DAG-satisfying* iff any DAG (directed acyclic graph) labelled only by relations containing b is satisfiable. □

Now, we shall classify the DAG-satisfying relations, after an auxiliary definition.

Definition 4.10 (Source) Let G be a DAG. Then a node v in G is said to be a *source* iff there are no arcs which end in v. □

Proposition 4.11 The basic relations $\prec$, d, o, f, s and $\equiv$, and their respective converses, are DAG-satisfying.

Proof: We show that any DAG labelled only by relations containing a fixed basic relation b, when b is one of the above relations, is satisfiable with some model M. Indeed, we prove the stronger result[2] that we can choose the satisfying M such that

- when b is d or o, all intervals overlap at some open interval
- when b is f, every interval has the same right ending point
- when b is s, every interval has the same left ending point
- when b is $\equiv$, all intervals are identical.

The result for the converse relations follows by an analogous construction.

Induction on the number of nodes in the DAG G. The case when $n = 0$ is trivial. Suppose that G has $n+1$ elements, and remove a source g from G.

[2]When b is $\prec$, we need not strengthen the result.

By induction, the remaining graph G' is satisfiable by a model M satisfying the required condition for the relation b. We shall now construct a model M' of G, which agrees with M on every interval variable in G'. The satisfying interval, denoted s, for the remaining interval variable represented by the node g, is chosen as follows, depending on b and M. Note that M satisfies the above conditions.

- When b is $\prec$, choose s to be any interval strictly before every interval in M
- When b is d, choose s to be an interval which is within the common open interval of the intervals in M
- When b is o, choose s to have its left ending point to the left of every interval in M, and its right ending point to be in the middle of the common interval of the intervals in M
- When b is f, choose s to have the same right ending point as the intervals in M, and the left ending point to be in the middle of the interval in M which has the rightmost left ending point
- When b is s, choose s to have the same left ending point as the intervals in M, and the right ending point to be in the middle of the interval in M which has the leftmost right ending point
- When b is $\equiv$, choose s to be identical to the intervals in M.

Obviously, M' is a model of G satisfying the requirements. $\Box$

We may note that m is not DAG-satisfying: take interval variables I_1, I_2 and I_3 related by $I_1(\mathsf{m})I_2$, $I_2(\mathsf{m})I_3$ and $I_1(\mathsf{m})I_3$. This is a DAG which is not satisfiable.

5 Tractable Algebras

Now we define the algebras which are to be analysed[3].

[3]This definition differs slightly from that of [Drakengren and Jonsson, 1996], but is easily verified (using Nebel and Bürckert's software [1993]) to result in the same maximal tractable algebras.

Definition 5.1 (The subclasses $A(r,b)$) Let b be a DAG-satisfying basic relation and r an acyclic relation containing b. First define the subclasses $A_1(b)$, $A_2(r,b)$ and $A_3(r,b)$ by

$$A_1(b) = \{r' \cup (b\; b^{\smile}) | r' \in \mathcal{A}\},$$

$$A_2(r,b) = \{r' \cup (b) | r' \subseteq r\}$$

and

$$A_3(r,b) = \{r' \cup (\equiv) | r' \in A_2(r,b)\} \cup \{(\equiv)\}.$$

Then set

$$B = A_1(b) \cup A_2(r,b) \cup A_3(r,b)$$

and finally define the subclass $A(r,b)$ by

$$A(r,b) = B \cup \{x^{\smile} | x \in B\} \cup \{(\;)\}.$$

$\square$

Corollary 5.2 Let r be an acyclic relation, $r' \subseteq r$, and b be some DAG-satisfying basic relation. Then $A(r',b) \subseteq A(r,b)$.
Proof: By the construction of $A(r,b)$. $\square$

Thus, by Corollary 5.2 and Corollary 4.3, it is sufficient to use maximal acyclic relations when constructing the algebras $A(r,b)$.

Now, using Proposition 4.8 we can construct twenty $A(r,b)$, by choosing r to be one of the maximal acyclic relations above, and choosing b to be an element in the chosen r except for m or $\mathsf{m}^{\smile}$. The reason why we get only twenty combinations (and not forty) is that the algebras are closed under the converse operation. Note that this exhausts the choices of parameters in $A(\cdot,\cdot)$, by Corollary 5.2 and Proposition 4.8.

We now prepare for an explicit listing of the algebras.

Proposition 5.3 Let r be maximal acyclic, $b \in r$ and $b \notin \{\mathsf{m}, \mathsf{m}^{\smile}\}$. Then $A(r,b)$ is the set of all $r' \in \mathcal{A}$ satisfying one of the following inclusions:

$$\begin{array}{llllll}
(b\; b^{\smile}) & \subseteq & r' & & \\
(b) & \subseteq & r' & \subseteq & (\equiv) \cup r \\
(b^{\smile}) & \subseteq & r' & \subseteq & (\equiv) \cup r^{\smile} \\
 & & r' & \subseteq & (\equiv).
\end{array}$$

Proof: An easy comparison with the definitions. □

Using this, we can list the algebras in a reasonably compact way. See the appendix for a complete listing of them all.

Proposition 5.4 Each of the $A(r,b)$ sets are algebras containing 2178 elements, and each contains exacly three basic relations, namely $\equiv$, b and $b^{\smile}$. Furthermore, all of these twenty algebras are distinct.
Proof: That the sets are algebras (i.e. closed under converse, intersection and composition) is verified by running the utility `aclose` by Nebel and Bürckert [1993]. The sizes of the algebras, distinctness, and what basic relations are included, can easily be obtained from the explicit listings above. □

We have four algebras $A(r,\prec)$, all containing the relations $(\equiv)$, $(\prec)$, $(\prec\,\equiv)$, $(\succ)$, $(\succ\,\equiv)$ and $(\prec\,\succ)$, expressing the notion of *sequentiality*, which is useful for solving reasoning problems under the assumption that actions always occur in sequence [Sandewall, 1994]. Note that the ORD-Horn algebra does *not* contain the relation $(\prec\,\succ)$, and thus cannot express sequentiality.

We now state the algorithm which we shall show in Theorem 5.9 solves satisfiability for these algebras, after a short definition.

Definition 5.5 (Strong component) A subgraph C of a graph G is said to be a *strong component* of G iff it is maximal such that for any nodes a, b in C, there is always a path in G from a to b. □

In fact, this algorithm is very similar to that of van Beek [1992], improved and used by Gerevini *et al.* [1993], but here used on intervals instead of points. In Section 6, we will show how the algorithm runs on an example.

We now state a simple result which holds for directed graphs in general.

Proposition 5.7 Let G be loop-free[4] with an acyclic subgraph D. Then those arcs of G which are not in D can be reoriented so that the resulting graph is acyclic.
Proof: Induction over the number n of nodes in G that are not in D. For $n=0$, the result is trivial. So, suppose that there are $n+1$ nodes in G that are not in D, and remove an arbitrary node v of these, resulting in the graph G'. By induction, the arcs of G' can be reoriented to form a DAG G''. Now

[4]A graph is said to be loop-free if it has no arcs from a node v to the node v.

Algorithm 5.6 ($\mathit{ISAT}(A(r,b))$)

```
1  Redirect the arcs of G so that all relations
   are in A1(b) ∪ A2(r, b) ∪ A3(r, b)
2  Let G' be the graph obtained from G by removing arcs which are not
   labelled by some relation in A2(r, b) ∪ A3(r, b)
3  Find all strong components C in G'
4  for every arc e in G whose relation
      does not contain ≡
5     if e connects two nodes in some C then
6        reject
7     endif
8  endfor
9  accept
```

□

add the node v to G'' obtaining G''', and reorient any arcs between G'' and v (in either direction) towards v. Since the graph is loop-free, no cycles are added by this operation, so G''' is acyclic. □

We now specialise this result.

Corollary 5.8 Let G be loop-free with an acyclic subgraph D, b a DAG-satisfying basic relation, and let the arcs of D be labelled by relations containing b, and the arcs not in D be labelled by relations containing both b and $b^{\smile}$. Then G is satisfiable.

Proof: Reorient the arcs of G like in Proposition 5.7, yielding a DAG G'. In this construction, whenever an arc is reoriented, also invert the relation on that arc, so that G' is satisfiable iff G is. By the construction, only arcs containing *both* b and $b^{\smile}$ have been reoriented, so every arc in the DAG G' contains b and, thus, since b is DAG-satisfying, G' is satisfiable, and consequently, also G is satisfiable. □

Theorem 5.9 Algorithm 5.6 correctly solves satisfiability for $A(r,b)$.

Proof: First, note that line 1 does not change satisfiability of G and can always be done according to the definition of $A(r,b)$.

Suppose that the algorithm rejects. Then there is an arc e not containing $\equiv$ that connects two nodes within a strong component C in G'. Suppose that $e \in A_2(r,b)$. Then there is a cycle of relations in $A_2(r,b) \cup A_3(r,b)$ containing e, and by Proposition 4.6, since e does not contain $\equiv$, C is unsatisfiable. Thus, suppose that $e \notin A_2(r,b)$ and that C does not contain any relation from $A_2(r,b)$. Then by Proposition 4.6 all relations in C have to be satisfied as $\equiv$. But since e does not contain $\equiv$, unsatisfiability results.

Now suppose that the algorithm accepts. Thus every strong component C can be collapsed to one interval, removing all arcs which would start and end in the collapsed interval, retaining the same condition for satisfiability, using the same argument as above. After the collapsing, the subgraph obtained by considering only arcs labelled by relations in $A_2(r,b) \cup A_3(r,b)$ will be acyclic. Since by construction every relation in $A_2(r,b) \cup A_3(r,b)$ contains the relation b, and the remaining arcs are labelled by relations containing both b and $b^\smile$, the graph is satisfiable by Corollary 5.8 (note that the graph will be loop-free, since every node is contained in some strong component). $\Box$

Theorem 5.10 Algorithm 5.6 runs in linear time in the size of G.
Proof: Line 1 is easily done in linear time. Strong components can be found in linear time (see e.g. Baase [1988]). Also the final test can be done in linear time. $\Box$

Corollary 5.11 Satisfiability of $A(r,b)$ is solvable in linear time.
Proof: From Theorem 5.9 and Theorem 5.10. $\Box$

Next, for the maximality results.

Proposition 5.12 The eight algebras $A(r,b)$ which have $b \in \{\mathsf{f}, \mathsf{s}\}$ are maximal tractable algebras.
Proof: By running the utility `atry` [Nebel and Bürckert, 1993], which generates minimal extensions of subclasses by adding a relation and computing the closure of that class. For these algebras, no nontrivial extensions were found (i.e. every extension results in $\mathcal{A}$), and since $ISAT(\mathcal{A})$ is NP-complete by Proposition 2.2, the result follows by Proposition 3.2. $\Box$

The remaining algebras are not maximal, as shown by Drakengren and Jonsson [1997a], where each is extended with 134 elements with retained tractability. However, the present algorithm has a lower asymptotic complexity than the one that is needed for the extended algebras.

Finally, we cover a case which is related to the $A(\cdot,\cdot)$ algebras, but occurs when every relation contains $\equiv$.

Definition 5.13 (The algebra $A_{\equiv}$) Define the algebra $A_{\equiv}$ to contain every relation that contains $\equiv$, and the empty relation $(\ )$. It is easy to see that $A_{\equiv}$ contains 4097 elements. □

For this algebra, we have the following trivial algorithm.

Algorithm 5.14 (Satisfiability in $A_{\equiv}$)

```
if some arc is labelled by ( ) then
    reject
else
    accept
endif
```

□

Proposition 5.15 Algorithm 5.14 correctly solves satisfiability in $A_{\equiv}$ in linear time. Furthermore, it is a maximal tractable subclass of $\mathcal{A}$.
Proof: Correctness and complexity results are trivial. The maximality follows by running the utility `atry` [Nebel and Bürckert, 1993], which generates no nontrivial extensions of the algebra. □

The algebra $A_{\equiv}$ certainly raises doubts about whether the *size* of a subalgebra can be used to judge its usefulness, since its expressivity is obviously too weak to be of any use.

6 Example

In this section we show how to model an industrially-inspired problem using one of the algebras, and exemplify how the algorithm runs on this problem.

Consider the following fragment of a chemical process. First denote different chemical substances by letters $u, v, \ldots, z$ possibly with primes on them. Then we have seven subprocesses $P_1, \ldots, P_7$, operating as follows.

P_1 melts x, producing x',
P_2 melts y and adds u', producing y',
P_3 melts z, producing z',
P_4 heats y', producing y'',
P_5 mixes x', y'', z' and w', producing x'',
P_6 pulverises u, producing u', and
P_7 cools w, and u' is added, producing w'.

Initially, all "unprimed" chemicals are available, whereas the "primed" ones are to be produced. The process itself, together with its controllers, impose the following ordering restrictions. The processes $P_1, \ldots, P_5$ are to be scheduled in sequence, P_5 necessarily finishing the sequence. Furthermore P_2 has to be scheduled strictly before P_4. P_6 must be completed before or just when P_2 starts, and P_7 is to take place either before or during the execution of P_4. The reason for this is that P_4 is a power-intensive process which causes transients in the power supply for process P_7 just when P_4 is started, and since temperature is critical during this cooling process, this cannot be allowed. Furthermore, P_6 has to start before P_7 starts, and P_6 must not finish before P_6 starts.

In order to optimise the ordering of processes in context with the rest of the production (which is not included here), we would like to allow any ordering consistent with the above constraints. However, the last requirement of P_6 not fininshing before P_6 starts, is not supported by the existing controllers so we would like to verify that this always holds, given all the other constraints, since otherwise, we would need to upgrade the controller.

The example is formalised by letting processes P_i represent interval variables. We obtain the following Allen relations.

$$\begin{array}{llllll}
P_1 & (\prec\ \succ) & P_2, & P_1 & (\prec\ \succ) & P_3, \\
P_1 & (\prec\ \succ) & P_4, & P_2 & (\prec\ \succ) & P_3, \\
P_2 & (\prec\ \succ) & P_4, & P_3 & (\prec\ \succ) & P_4, \\
P_2 & (\prec) & P_4, & P_i & (\prec) & P_5 \text{ for all } i \neq 5, \\
P_6 & (\prec\ \mathsf{m}) & P_2, & P_7 & (\prec\ \mathsf{d}) & P_4.
\end{array}$$

The condition we would like to verify, is that the constraint

$$P_6(\mathsf{o\ m})P_7$$

is satisfied in every model. Note that this is equivalent to checking the unsatisfiability of the above constraints together with the constraint

$$P_6(\equiv\ \prec\ \succ\ \ \mathsf{d}\ \mathsf{d}^{\smile}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}\ \mathsf{s}^{\smile}\ \mathsf{f}\ \mathsf{f}^{\smile})P_7,$$

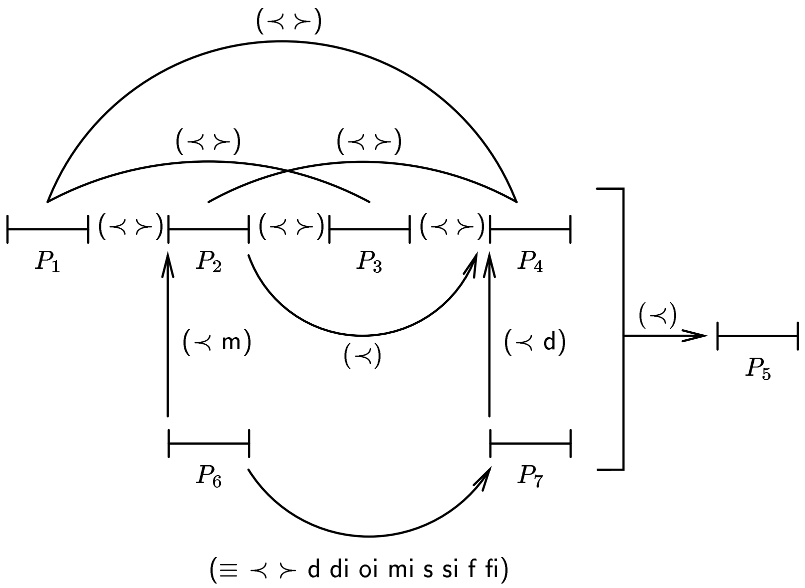

Figure 1: The network describing the constraints on the process.

the complement of the relation (o m), and that all constraints in that network are included in e.g. the tractable algebra $A(m_3, \prec)$ of Proposition 1.1.

We can now show how Algorithm 5.6 performs on this set of constraints (depicted in Figure 1). First, all arcs are already contained in the set

$$A_1(\prec) \cup A_2(m_3, \prec) \cup A_3(m_3, \prec),$$

so nothing needs to be done in the first step. For the second step, the graph G' can be represented by the following set of relations.

$$\begin{array}{llllll} P_2 & (\prec) & P_4, & P_6 & (\prec\ \mathsf{m}) & P_2, \\ P_7 & (\prec\ \mathsf{d}) & P_4, & P_i & (\prec) & P_5 \text{ for all } i \neq 5. \end{array}$$

For the third step, we find all strong components in this graph. It is easy to see that there are no cycles in G', so the set of strong components will be identical to the set of nodes in the graph. Next, we need to check if some relation in the original set of constraints connects two nodes in some component, that is, it is a *loop*, in this case. Obviously, there are no such loops here. Thus, the algorithm accepts, and by correctness, the constraints are satisfiable, and we have no guarantee that our requirement on the process always holds. Consequently, the controller had better be upgraded.

7 The Applicability of Path Consistency

The first attempts at reasoning in Allen's algebra used the *path consistency algorithm* [Montanari, 1974; Mackworth, 1977]; in particular, Allen used such an algorithm [Allen, 1983] as a sound method for checking consistency of interval networks. Also, a recent attempt by Ladkin and Reinefeld [1992] uses the path consistency algorithm as a subroutine in a backtracking search algorithm for reasoning in the full algebra. At each branching step, the algorithm splits disjunctive relations into relations from some algebra for which the path consistency algorithm is complete, thus reducing its branching factor depending on the size of the algebra used. Nebel [1996] proved this method to be correct, and evaluated its efficiency using the ORD-Horn algebra [Nebel and Bürckert, 1995], for which the path consistency algorithm is complete.

In order to make the algebras of this paper useful in such contexts, we prove that the path consistency algorithm is complete for them all. Note, however, that this does not represent an improvement wrt. complexity when using the algebras separately, since path-consistency algorithms typically require at least $O(n^3)$ time [Vilain *et al.*, 1989] where n is the number of interval variables, to be compared to the result of Theorem 5.10, which represents a better running time in the general case.

Definition 7.1 (Path consistent network) Let $V = \{v_1, \ldots, v_m\}$ be interval variables, and $R = \{r_{ij} | 1 \leq i, j \leq m\}$ Allen relations, where r_{ij} holds between v_i and v_j. Then the network $G = \langle V, R \rangle$ is said to be *path consistent* iff for all v_i, v_j and v_k, it holds that $v_i \circ v_j \supseteq v_k$. □

Definition 7.2 (Path consistency algorithm) Let $V = \{v_1, \ldots, v_m\}$ be interval variables, and $R = \{r_{ij} | 1 \leq i, j \leq m\}$ Allen relations, where r_{ij} holds between v_i and v_j. The *path consistency algorithm* essentially computes

$$r_{ij} \leftarrow r_{ij} \cap (r_{ik} \circ r_{kj})$$

for all i, j, k until no more changes occur[5]. □

The algorithm can be implemented in a variety of ways, but they are all equivalent to the above definition in terms of input-output behaviour.

[5]The $\leftarrow$ represents *assignment* of r_{ij}.

Corollary 7.3 Let $V = \{v_1, \ldots, v_m\}$ be interval variables, and

$$R = \{r_{ij} | 1 \leq i, j \leq m\}$$

a set of Allen relations, where r_{ij} is the relation between v_i and v_j. Then the result of the path consistency algorithm running on $\langle V, R \rangle$ is path consistent.
Proof: By Definition 7.1 and Definition 7.2. $\square$

Definition 7.4 (Deciding satisfiability) We say that the path consistency algorithm *decides satisfiability* for a subclass A of Allen's algebra iff for any network $G = \langle V, R \rangle$ of relations from A, if the path consistency algorithm on G is run on G producing the network G', then G is satisfiable iff G' does not contain the relation ( ). $\square$

Corollary 7.5 The path consistency algorithm decides satisfiability for a subalgebra A of Allen's algebra iff every path consistent network of relations from A not containing ( ) is satisfiable.
Proof: Let $G = \langle V, R \rangle$ be a network of relations from A, and run the path consistency algorithm on G, producing the network G'. Since A is an algebra, G', too, contains only relations from A, by the operation of the algorithm. Now, if G is satisfiable, G' cannot contain ( ), since the algorithm only makes implicit relations explicit (i.e. it is *sound*). If G' does not contain ( ), then G' is satisfiable by the condition, and since G' is more constrained than G, G is also satisfiable. $\square$

We start by a trivial observation.

Proposition 7.6 The path consistency algorithm decides satisfiability for $A_{\equiv}$.
Proof: If a path consistent network with relations from $A_{\equiv}$ does not contain ( ), then it is satisfiable by setting all intervals equal. The result follows by Corollary 7.5. $\square$

In order to show the result for the remaining algebras, we first need the following lemmata.

Lemma 7.7 Let $G = \langle V, R \rangle$ be a path consistent network. Then G cannot contain a cycle S with relations only from the set $A = A_2(r, b) \cup A_3(r, b)$, and at least one relation e from $A_2(r, b)$.

Proof: Without loss of generality, suppose that r is one of the four explicitly stated maximal acyclic relations of Proposition 4.8 (i.e. exclude their converses), and assume that such a cycle S exists. Let the interval variables in S be $s_1, s_2, \ldots, s_n$ with relations named $s_i r_{ij} s_j$ for $1 \leq i, j \leq n$. By the construction of A, we have one of the following cases:

1. For any $r \in A$, if IrJ holds in some model M, then $I^- \leq J^-$, and if IeJ holds in some model M, then $I^- < J^-$ holds in M.

2. For any $r \in A$, if IrJ holds in some model M, then $I^+ \leq J^+$, and if IeJ holds in some model M, then $I^+ < J^+$ holds in M.

By symmetry, it is enough to prove the result for the first case.

From path consistency, we get that for any $1 \leq i < k < j \leq n$, $r_{ij} \subseteq r_{ik} \circ r_{kj}$. It is easy to see that by property 1 above, if $r, r' \in A$ and if $Ir \circ r'J$ holds in some model M, then $I^- \leq J^-$ holds in M, and further, by a simple induction and the path consistency property, that for any $1 \leq i < j \leq n$, if $s_i r_{ij} s_j$ holds in some model M, then $s_i^- \leq s_j^-$ holds in M. Suppose, by renumbering the cycle nodes, that $e = r_{n1}$. Thus by property 1 it has to hold that $s_n^- < s_1^-$ in any model of G. But by the above we have that $s_1^- \leq s_n^-$, which is a contradiction. □

Lemma 7.8 Let $G = \langle V, R \rangle$ be a path consistent network. Then G cannot contain a cycle S with relations only from the set $A = A_3(r, b)$, where two nodes in S are connected by some relation e not containing $(\equiv)$.

Proof: Without loss of generality, suppose that r is one of the four explicitly stated maximal acyclic relations of Proposition 4.8 (i.e. exclude their converses), and assume that such a cycle S exists. Let the interval variables in S be $s_1, s_2, \ldots, s_n$ with relations named $s_i r_{ij} s_j$ for $1 \leq i, j \leq n$. By the construction of A, we have one of the following cases:

1. For any $r \in A$, if IrJ holds in some model M, then $I^- \leq J^-$ and $I^+ \geq J^+$, and if IeJ holds in some model M, then I and J are not identical in M.

2. For any $r \in A$, if IrJ holds in some model M, then $I^+ \leq J^+$ and $I^- \geq J^-$, and if IeJ holds in some model M, then I and J are not identical in M.

By symmetry, it is enough to prove the result for the first case.

From path consistency, we get that for any $1 \leq i < k < j \leq n$, $r_{ij} \subseteq r_{ik} \circ r_{kj}$. It is easy to see that by property 1 above, if $r, r' \in A$ and if

$Ir \circ r'J$ holds in some model M, then $I^- \leq J^-$ and $I^+ \geq J^+$ hold in M, and further, by a simple induction and the path consistency property, that for any $1 \leq i, j \leq n$, if $s_i r_{ij} s_j$ holds in some model M, then $s_i^- \leq s_j^-$ and $s_i^+ \geq s_j^+$ hold in M. But this implies that $r_{ij} \subseteq (\equiv)$ for every relation r_{ij} in the cycle. Let the relation e connect nodes a and b. Now, since e does not contain $\equiv$, $r_{ij} = (\ )$ by path consistency, contradicting the existence of a cycle with the required properties. □

The main result of the section is proved next.

Theorem 7.9 Let $A = A(r, b)$ be one of the algebras from Definition 5.1. Then the path consistency algorithm decides consistency for A.
Proof: Again, we use Corollary 7.5, so suppose that $G = \langle V, R \rangle$ is a path-consistent network of relations from A, not containing $(\ )$, and that G is not satisfiable. Without loss of generality, assume that G only contains relations from $A(b) \cup A_2(r, b) \cup A_3(r, b)$, and consider the graph G', obtained from G by removing arcs which are not labelled by some relation in $A_2(r, b) \cup A_3(r, b)$. By the correctness of Algorithm 5.6, there exists a strong component C of G' and an arc e of G not containing $\equiv$ such that e connects two nodes in C. Suppose that $e \in A_2(r, b)$. Then there is a cycle S of relations in $A_2(r, b) \cup A_3(r, b)$ containing e, which is unsatisfiable. But this is impossible by Lemma 7.7. Thus, suppose that $e \notin A_2(r, b)$ and that C does not contain any relation from $A_2(r, b)$. Then there is a cycle S of relations in $A_3(r, b)$ where two nodes are connected by e. But this is impossible by Lemma 7.8. The result follows by Corollary 7.5. □

8 Discussion

Nebel and Bürckert [1995] argue that the ORD-Horn algebra is an improvement in quantitative terms over previous approaches, since it covers more than 10 percent of the full algebra. Certainly this is a valid argument only because the ORD-Horn algebra *includes* the previous algebras; otherwise we have a counterexample in the $A_\equiv$ algebra, which is much larger than the ORD-Horn algebra, but is clearly of no use. We may mention that the 21 algebras of this paper together cover about 92 percent of $\mathcal{A}$, and that there are only two relations in the ORD-Horn algebra which are not elements of any of the algebras: (m) and $(\mathsf{m}^\smile)$. From a cognitive perspective, the exclusion of these relations is not a serious restriction, as Freksa [1992] notes, since they are not likely to occur in any context reasoning about, for instance, perception of the physical world.

It is also argued by Nebel and Bürckert [1995] that a useful algebra should contain all the basic relations, since otherwise, complete knowledge cannot be specified. However, since the unique maximality of the ORD-Horn class shows that there exists no tractable subalgebra which contains both all the basic relations and the relations expressing *sequentiality* (notably the $(\prec \succ)$ relation), this argument fails. Furthermore, four of our algebras can indeed express this sequentiality requirement, which underlies many systems (see e.g. [Sandewall, 1994]).

Given four such algebras, a justified question is whether these are just four out of, for example, four hundred such tractable classes, that is, what makes these algebras relevant? Fortunately, recent research [Drakengren and Jonsson, 1997b] has shown that the two maximal tractable extensions of these four algebras found by Drakengren and Jonsson [1997a] are the *only* maximal tractable algebras capable of expressing sequentiality. Similarly, in the point-interval algebra, we have only five maximal tractable subclasses [Jonsson *et al.*, 1996] indicating that maximal tractable algebras are typically scarce.

Also, as a long-term goal, it would be useful to classify *all* maximal tractable subalgebras of the full algebra, since then an application using networks of interval relations could search for the best algebra to use, or otherwise report that no such algebra exists. Since there are 2^{8192} subsets of the full algebra, the task is clearly nontrivial, even using computer-supported proof methods. Our work with Bäckström on the point-interval algebra [Jonsson *et al.*, 1996] and our recent partial classification of the Allen algebra [Drakengren and Jonsson, 1997b] show that a complete classification might not be out of reach.

Recently, a proof method for the Allen algebra has been presented by Ligozat [1996], which was applied to find a new tractability proof for satisfiability in the ORD-Horn algebra. Even though the method seems to be quite general, it is not applicable to the algebras of this paper, since the *convexity* properties which are crucial are not satisfied, notably by relations like $(\prec \succ)$.

Finally, it is interesting to note that although we have an algorithm for satisfiability of all the algebras, we cannot automatically conclude that the problem of *entailment*, which is that of deciding what relation is induced between two intervals, is solved: in the case when a tractable algebra contains all of the basic relations, this problem reduces to satisfiability, as shown by Golumbic and Shamir [1993], but this is not at all obvious when that restriction is not satisfied.

9 Conclusions

We have identified 21 new large tractable fragments of Allen's interval algebra, of which nine have been proved maximal tractable. Further, we have presented a linear time algorithm for deciding satisfiability of these. In addition, all the algebras are considerably larger (in quantity) than the ORD-Horn subalgebra, but thus cannot contain all the basic relations. Also, four of the algebras can express the relations $(\equiv)$, $(\prec)$, $(\prec \equiv)$, $(\succ)$, $(\succ \equiv)$ and $(\prec \succ)$ (in addition to the "nonrelation"), which is necessary and sufficient for expressing the notion of sequentiality. Finally, we showed that the path consistency algorithm decides satisfiability for all of the algebras.

References

[AAAI '96, 1996] American Association for Artificial Intelligence. *Proceedings of the 13th (US) National Conference on Artificial Intelligence (AAAI '96)*, Portland, OR, USA, August 1996.

[Allen, 1983] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[Baase, 1988] Sara Baase. *Computer Algorithms.* Addison-Wesley Publishing Company, 1988.

[Drakengren and Jonsson, 1996] Thomas Drakengren and Peter Jonsson. Maximal tractable subclasses of Allen's interval algebra: Preliminary report. In AAAI '96 [1996], pages 389–394.

[Drakengren and Jonsson, 1997a] Thomas Drakengren and Peter Jonsson. Eight maximal tractable subclasses of Allen's algebra with metric time. *Journal of Artificial Intelligence Research*, 7:25–45, 1997.

[Drakengren and Jonsson, 1997b] Thomas Drakengren and Peter Jonsson. Towards a complete classification of tractability in Allen's algebra. In Martha E. Pollack, editor, *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI '97)*, Nagoya, Japan, August 1997. Morgan Kaufmann.

[Freksa, 1992] Christian Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54:199–227, 1992.

[Gerevini *et al.*, 1993] Alfonso Gerevini, Lenhart Schubert, and Stephanie Schaeffer. Temporal reasoning in Timegraph I–II. *SIGART Bulletin*, 4(3):21–25, 1993.

[Golumbic and Shamir, 1993] Martin Charles Golumbic and Ron Shamir. Complexity and algorithms for reasoning about time: A graph-theoretic approach. *Journal of the ACM*, 40(5):1108–1133, 1993.

[Jonsson *et al.*, 1996] Peter Jonsson, Thomas Drakengren, and Christer Bäckström. Tractable subclasses of the point-interval algebra: A complete classification. In J. Doyle and L. Aiello, editors, *Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning (KR '96)*, pages 352–363, Cambridge, MA, USA, October 1996. Morgan Kaufmann.

[Ladkin and Reinefeld, 1992] Peter B. Ladkin and Alexander Reinefeld. Effective solution of qualitative interval constraint problems. *Artificial Intelligence*, 57(1):105–124, 1992.

[Ligozat, 1996] Gérard Ligozat. A new proof of tractability for ORD-Horn relations. In AAAI '96 [1996], pages 395–401.

[Mackworth, 1977] Alan K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99–118, 1977.

[Montanari, 1974] Ugo Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7:95–132, 1974.

[Nebel and Bürckert, 1993] Bernhard Nebel and Hans-Jürgen Bürckert. Software for machine assisted analysis of Allen's interval algebra. Available from the authors by anonymous ftp from `duck.dfki.uni-sb.de` as `/pub/papers/DFKI-others/RR-93-11.programs.tar.Z`, 1993.

[Nebel and Bürckert, 1995] Bernhard Nebel and Hans-Jürgen Bürckert. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *Journal of the ACM*, 42(1):43–66, 1995.

[Nebel, 1996] Bernhard Nebel. Solving hard qualitative temporal reasoning problems: Evaluating the efficiency of using the ORD-Horn class. In Wolfgang Wahlster, editor, *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI '96)*, Budapest, Hungary, August 1996. Wiley.

[Sandewall, 1994] Erik Sandewall. *Features and Fluents.* Oxford University Press, 1994.

[van Beek, 1989] Peter van Beek. Approximation algorithms for temporal reasoning. In N. S. Sridharan, editor, *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI '89)*, pages 1291–1296, Detroit, MI, USA, August 1989. Morgan Kaufmann.

[van Beek, 1992] Peter van Beek. Reasoning about qualitative temporal information. *Artificial Intelligence*, 58:297–326, 1992.

[Vilain *et al.*, 1989] Marc B. Vilain, Henry A. Kautz, and Peter G. van Beek. Constraint propagation algorithms for temporal reasoning: A revised report. In *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan Kaufmann, San Mateo, CA, 1989.

Appendix

1 An Explicit Listing of the Algebras

Proposition 1.1 First abbreviate the maximal acyclic relations by

$$\begin{array}{lcl} m_1 & = & (\prec\ \mathsf{d}^\smile\ \mathsf{o\ m\ s\ f}^\smile) \\ m_2 & = & (\prec\ \mathsf{d}^\smile\ \mathsf{o\ m\ s}^\smile\ \mathsf{f}^\smile) \\ m_3 & = & (\prec\ \mathsf{d\ o\ m\ s\ f}) \\ m_4 & = & (\prec\ \mathsf{d\ o\ m\ s\ f}^\smile), \end{array}$$

obtaining the remaining ones by taking the converses of these.

Now $A(m_1, \prec)$ is defined by

$$\begin{array}{lcccl} (\prec\ \succ) & \subseteq & r' & & \\ (\prec) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \mathsf{d}^\smile\ \mathsf{o\ m\ s\ f}^\smile) \\ (\succ) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \mathsf{d\ o}^\smile\ \mathsf{m}^\smile\ \mathsf{s}^\smile\ \mathsf{f}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_2, \prec)$ by

$$\begin{array}{lcccl} (\prec\ \succ) & \subseteq & r' & & \\ (\prec) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \mathsf{d}^\smile\ \mathsf{o\ m\ s}^\smile\ \mathsf{f}^\smile) \\ (\succ) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \mathsf{d\ o}^\smile\ \mathsf{m}^\smile\ \mathsf{s\ f}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_3, \prec)$ by

$$\begin{array}{lcccl} (\prec\ \succ) & \subseteq & r' & & \\ (\prec) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \mathsf{d\ o\ m\ s\ f}) \\ (\succ) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \mathsf{d}^\smile\ \mathsf{o}^\smile\ \mathsf{m}^\smile\ \mathsf{s}^\smile\ \mathsf{f}^\smile) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_4, \prec)$ by

$$\begin{array}{lcccl} (\prec\ \succ) & \subseteq & r' & & \\ (\prec) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \mathsf{d\ o\ m\ s\ f}^\smile) \\ (\succ) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \mathsf{d}^\smile\ \mathsf{o}^\smile\ \mathsf{m}^\smile\ \mathsf{s}^\smile\ \mathsf{f}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_1{}^\smile, \mathsf{d})$ by

$$\begin{array}{llllll}(\mathsf{d}\ \mathsf{d}^{\smile}) & \subseteq & r' & & \\ (\mathsf{d}) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \ \mathsf{d}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}^{\smile}\ \mathsf{f}) \\ (\mathsf{d}^{\smile}) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \ \mathsf{d}^{\smile}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}\ \mathsf{f}^{\smile}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_2{}^{\smile}, \mathsf{d})$ by

$$\begin{array}{llllll}(\mathsf{d}\ \mathsf{d}^{\smile}) & \subseteq & r' & & \\ (\mathsf{d}) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \ \mathsf{d}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}\ \mathsf{f}) \\ (\mathsf{d}^{\smile}) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \ \mathsf{d}^{\smile}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}^{\smile}\ \mathsf{f}^{\smile}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_3, \mathsf{d})$ by

$$\begin{array}{llllll}(\mathsf{d}\ \mathsf{d}^{\smile}) & \subseteq & r' & & \\ (\mathsf{d}) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \ \mathsf{d}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}\ \mathsf{f}) \\ (\mathsf{d}^{\smile}) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \ \mathsf{d}^{\smile}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}^{\smile}\ \mathsf{f}^{\smile}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_4, \mathsf{d})$ by

$$\begin{array}{llllll}(\mathsf{d}\ \mathsf{d}^{\smile}) & \subseteq & r' & & \\ (\mathsf{d}) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \ \mathsf{d}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}\ \mathsf{f}^{\smile}) \\ (\mathsf{d}^{\smile}) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \ \mathsf{d}^{\smile}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}^{\smile}\ \mathsf{f}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_1, \mathsf{o})$ by

$$\begin{array}{llllll}(\mathsf{o}\ \mathsf{o}^{\smile}) & \subseteq & r' & & \\ (\mathsf{o}) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \ \mathsf{d}^{\smile}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}\ \mathsf{f}^{\smile}) \\ (\mathsf{o}^{\smile}) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \ \mathsf{d}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}^{\smile}\ \mathsf{f}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_2, \mathsf{o})$ by

$$\begin{array}{llllll}(\mathsf{o}\ \mathsf{o}^{\smile}) & \subseteq & r' & & \\ (\mathsf{o}) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \ \mathsf{d}^{\smile}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}^{\smile}\ \mathsf{f}^{\smile}) \\ (\mathsf{o}^{\smile}) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \ \mathsf{d}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}\ \mathsf{f}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_3, \mathsf{o})$ by

$$\begin{array}{llllll} (\mathsf{o}\;\mathsf{o}^{\smile}) & \subseteq & r' & & \\ (\mathsf{o}) & \subseteq & r' & \subseteq & (\equiv \prec\; \mathsf{d}\;\mathsf{o}\;\mathsf{m}\;\mathsf{s}\;\mathsf{f}) \\ (\mathsf{o}^{\smile}) & \subseteq & r' & \subseteq & (\equiv \succ\; \mathsf{d}^{\smile}\;\mathsf{o}^{\smile}\;\mathsf{m}^{\smile}\;\mathsf{s}^{\smile}\;\mathsf{f}^{\smile}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_4, \mathsf{o})$ by

$$\begin{array}{llllll} (\mathsf{o}\;\mathsf{o}^{\smile}) & \subseteq & r' & & \\ (\mathsf{o}) & \subseteq & r' & \subseteq & (\equiv \prec\; \mathsf{d}\;\mathsf{o}\;\mathsf{m}\;\mathsf{s}\;\mathsf{f}^{\smile}) \\ (\mathsf{o}^{\smile}) & \subseteq & r' & \subseteq & (\equiv \succ\; \mathsf{d}^{\smile}\;\mathsf{o}^{\smile}\;\mathsf{m}^{\smile}\;\mathsf{s}^{\smile}\;\mathsf{f}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_1, \mathsf{s})$ by

$$\begin{array}{llllll} (\mathsf{s}\;\mathsf{s}^{\smile}) & \subseteq & r' & & \\ (\mathsf{s}) & \subseteq & r' & \subseteq & (\equiv \prec\; \mathsf{d}^{\smile}\;\mathsf{o}\;\mathsf{m}\;\mathsf{s}\;\mathsf{f}^{\smile}) \\ (\mathsf{s}^{\smile}) & \subseteq & r' & \subseteq & (\equiv \succ\; \mathsf{d}\;\mathsf{o}^{\smile}\;\mathsf{m}^{\smile}\;\mathsf{s}^{\smile}\;\mathsf{f}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_2{}^{\smile}, \mathsf{s})$ by

$$\begin{array}{llllll} (\mathsf{s}\;\mathsf{s}^{\smile}) & \subseteq & r' & & \\ (\mathsf{s}) & \subseteq & r' & \subseteq & (\equiv \succ\; \mathsf{d}\;\mathsf{o}^{\smile}\;\mathsf{m}^{\smile}\;\mathsf{s}\;\mathsf{f}) \\ (\mathsf{s}^{\smile}) & \subseteq & r' & \subseteq & (\equiv \prec\; \mathsf{d}^{\smile}\;\mathsf{o}\;\mathsf{m}\;\mathsf{s}^{\smile}\;\mathsf{f}^{\smile}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_3, \mathsf{s})$ by

$$\begin{array}{llllll} (\mathsf{s}\;\mathsf{s}^{\smile}) & \subseteq & r' & & \\ (\mathsf{s}) & \subseteq & r' & \subseteq & (\equiv \prec\; \mathsf{d}\;\mathsf{o}\;\mathsf{m}\;\mathsf{s}\;\mathsf{f}) \\ (\mathsf{s}^{\smile}) & \subseteq & r' & \subseteq & (\equiv \succ\; \mathsf{d}^{\smile}\;\mathsf{o}^{\smile}\;\mathsf{m}^{\smile}\;\mathsf{s}^{\smile}\;\mathsf{f}^{\smile}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_4, \mathsf{s})$ by

$$\begin{array}{llllll} (\mathsf{s}\;\mathsf{s}^{\smile}) & \subseteq & r' & & \\ (\mathsf{s}) & \subseteq & r' & \subseteq & (\equiv \prec\; \mathsf{d}\;\mathsf{o}\;\mathsf{m}\;\mathsf{s}\;\mathsf{f}^{\smile}) \\ (\mathsf{s}^{\smile}) & \subseteq & r' & \subseteq & (\equiv \succ\; \mathsf{d}^{\smile}\;\mathsf{o}^{\smile}\;\mathsf{m}^{\smile}\;\mathsf{s}^{\smile}\;\mathsf{f}) \\ & & r' & \subseteq & (\equiv), \end{array}$$

$A(m_1{}^{\smile}, \mathsf{f})$ by

$$\begin{array}{lllll}(\mathsf{f}\ \mathsf{f}^{\smile}) & \subseteq & r' & & \\ (\mathsf{f}) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \mathsf{d}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}^{\smile}\ \mathsf{f}) \\ (\mathsf{f}^{\smile}) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \mathsf{d}^{\smile}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}\ \mathsf{f}^{\smile}) \\ & & r' & \subseteq & (\equiv),\end{array}$$

$A(m_2{}^{\smile}, \mathsf{f})$ by

$$\begin{array}{lllll}(\mathsf{f}\ \mathsf{f}^{\smile}) & \subseteq & r' & & \\ (\mathsf{f}) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \mathsf{d}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}\ \mathsf{f}) \\ (\mathsf{f}^{\smile}) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \mathsf{d}^{\smile}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}^{\smile}\ \mathsf{f}^{\smile}) \\ & & r' & \subseteq & (\equiv),\end{array}$$

$A(m_3, \mathsf{f})$ by

$$\begin{array}{lllll}(\mathsf{f}\ \mathsf{f}^{\smile}) & \subseteq & r' & & \\ (\mathsf{f}) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \mathsf{d}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}\ \mathsf{f}) \\ (\mathsf{f}^{\smile}) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \mathsf{d}^{\smile}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}^{\smile}\ \mathsf{f}^{\smile}) \\ & & r' & \subseteq & (\equiv),\end{array}$$

and finally, $A(m_4{}^{\smile}, \mathsf{f})$ by

$$\begin{array}{lllll}(\mathsf{f}\ \mathsf{f}^{\smile}) & \subseteq & r' & & \\ (\mathsf{f}) & \subseteq & r' & \subseteq & (\equiv\ \succ\ \mathsf{d}^{\smile}\ \mathsf{o}^{\smile}\ \mathsf{m}^{\smile}\ \mathsf{s}^{\smile}\ \mathsf{f}) \\ (\mathsf{f}^{\smile}) & \subseteq & r' & \subseteq & (\equiv\ \prec\ \mathsf{d}\ \mathsf{o}\ \mathsf{m}\ \mathsf{s}\ \mathsf{f}^{\smile}) \\ & & r' & \subseteq & (\equiv),\end{array}$$

The remaining combinations, which are redundant, are obtained by the equation

$$A(r, b) = A(r^{\smile}, b^{\smile}).$$

$\square$