An application-independent support system for integrated assumption-based temporal reasoning

Clemens Beckstein
Univ. Erlangen-Nürnberg, IMMD-8
Am Weichselgarten 9
D-91058 Erlangen, Germany
beckstein@informatik.uni-erlangen.de

Tim Geisler
Univ. München, Institut für Informatik
Wagmüllerstr. 23
D-80538 München, Germany
geisler@informatik.uni-muenchen.de

Abstract

Reason maintenance systems and temporal reasoning systems are among the most prominent application-independent support systems used for complex AI applications and there are many applications that need both support for logical and temporal reasoning. Apparently it is not enough to just provide two isolated support modules. The logical and temporal subsystems have to be coupled in the right way. We present a hybrid support system integrating assumption-based logical and temporal reasoning, give a formal characterization, and show what an efficient incremental implementation of the system would look like.

1 Assumption-based temporal reasoning

Suppose we want to mass produce some machines the details of which are not important here. As part of this, we have to plan how to manufacture the parts of the machines. These plans can then be used to search for a schedule that guarantees a cost effective production of the parts on a given set of machines.

When planning or executing plans, assumptions have to be made — about future world states, the availability of resources or the duration of actions. These assumptions have to be managed explicitly in order to be able to make or retract them as needed. Explicit representation and processing of assumptions can be accomplished with assumption-based reason maintenance systems as the ATMS [dK86]. On top of the ATMS there usually are systems like the PNMS (Plan Network Maintenance System) [BLS92] representing and maintaining non-linear plans with assumptions.

It is often impossible to separate and sequentialize planning and scheduling in applications due to the strong interactions between them. The isolated generation of an "optimal" plan can render it impossible to find the best schedule. When viewed separately, a global optimum can often be produced from two suboptimal partial solutions. Therefore planning and scheduling have to be considered together.

Temporal reasoning is necessary for scheduling as well as for planning, but the requirements are different. For planning, a lot of qualitative temporal information is needed because the executability of plans has to be ensured and propositions about the ordering of events in time have to be made. In order to produce a schedule, propositions about absolute time points and durations have to be made, which require metric temporal information. For scheduling, metric constraints are not absolute; they can be different under different assumptions.

This problem can be solved by making the assumptions underlying these constraints *explicit*, i.e. by going from temporal constraints to assumption-based temporal constraints. By reasoning with temporal rules, normal boolean propositions become derivable, and by reasoning with logical rules new temporal constraints can be computed. Therefore, an integration of the assumption-based processing of temporal constraints and the assumption-based processing of logical constraints is necessary.

The product of this integration can then be added to a problem solver as a module which acts both as a conventional reason maintenance system and a temporal reasoning system.

In the rest of this paper, we present how such a module can be constructed from the ATMS by merging it with a simple assumption-based temporal reasoning system.

2 A simple assumption-based temporal reasoning system

The STP: The starting point of our development was the Simple Temporal Problem (STP[1]) [DMP91], a very simple formalism for temporal reasoning with

[1] In the sequel we will use the abbreviation (A)STP both for the problem and the system that can solve (assumption-based) simple temporal problems.

metric constraints. The STP is a constraint satisfaction problem over a set $\mathcal{T}$ of variables with the range $\mathcal{D}$, given by a set of constraints $\mathcal{C}$ the elements of which all have the form

$$t_j - t_i \leq a$$

where $t_i, t_j \in \mathcal{T}$ and $a \in \mathcal{D}$. The elements of the domain can be interpreted as time points or time distances and the constraints express upper bounds for the (directed) distance of two time points. Implicitly for each variable t_i a constraint $t_0 - t_i \leq 0$ is introduced, where t_0 is a distinguished variable. A solution of an STP is an assignment of the variables from $\mathcal{T}$ satisfying all constraints $\mathcal{C}$.

With this kind of constraints a lot of interesting temporal relations can be expressed (like the duration of jobs, partial ordering of jobs or deadlines and other information of importance for scheduling).

An STP can be represented as a labeled directed graph. Each variable is represented by a node and each constraint $t_j - t_i \leq a$ is represented by an edge from t_i to t_j with the label a. The test for satisfiability as well as the solutions can be computed from the shortest paths between the node representing the variable t_0 and all other nodes in the graph. Hence, the Floyd-Warshall algorithm (cf. [AHU74, CLR90]), which has a time complexity of $O(n^3)$ where n is the number of variables, can be used to solve an STP. The algorithm computes, as a side effect, maximal directed distances between each pair of time points represented by a variable. These can be viewed as solutions to the constraints in the sense described above.

Extension to the ASTP: In a lot of applications temporal constraints cannot be expressed absolutely as in the STP. We therefore generalized the STP to an ASTP[1] (Assumption-based Simple Temporal Problem). In the ASTP, constraints are expressed relative to a set of assumptions. Let $\mathcal{A}$ denote the universe of all assumptions. An ASTP is a constraint satisfaction problem over a set $\mathcal{T}$ of variables with the range $\mathcal{D}$, given by a set of constraints $\mathcal{C}$ with elements of the form

$$U \vdash t_j - t_i \leq a$$

($t_i, t_j \in \mathcal{T}$ and $a \in \mathcal{D}$ and $U \subseteq \mathcal{A}$ is a set of assumptions). In the ASTP we also have for each variable the implicit constraint $\emptyset \vdash t_0 - t_i \leq 0$. The ASTP is apparently downward compatible to the STP — an STP-constraint of the form $t_j - t_i \leq a$ is represented in the ASTP as $\emptyset \vdash t_j - t_i \leq a$.

The solutions $\mathcal{P}(\mathcal{A}) \rightarrow (\mathcal{T} \rightarrow \mathcal{D})$ of an ASTP for $\mathcal{C}$ are all the functions which map sets of assumptions[2] $U \subseteq \mathcal{A}$ to assignments for the variables in $\mathcal{T}$ and satisfy all constraints $(U' \vdash t_j - t_i \leq a) \in \mathcal{C}$ where $U' \subseteq U$.

[2] $\mathcal{P}(A)$ denotes the power set of the set A.

As the Floyd-Warshall algorithm can be generalized to path problems in weighted directed graphs where the edge weights are elements of a closed semiring $(S, \oplus, \odot, \overline{0}, \overline{1})$ [AHU74], a suitable semiring has to be identified to get an algorithm for the solution of an ASTP. Path problems in closed semirings can be formalized as follows:

The label $\lambda(p)$ of a path $p = \langle v_1, \ldots, v_k \rangle$ in such a graph is defined as

$$\lambda(p) := \lambda(v_1, v_2) \odot \ldots \odot \lambda(v_{k-1}, v_k),$$

where the label $\lambda(u, v)$ of an edge (u, v) is their weight. If (u, v) is not an edge of the graph, then $\lambda(u, v) = \overline{0}$.

The generalized Floyd-Warshall algorithm computes the summary l_{uv} of all labels of all paths for all pairs (u, v) of nodes in the graph:

$$l_{uv} := \bigoplus_{p=\langle u,\ldots,v\rangle} \lambda(p)$$

It can be sketched as follows:

```
for i,j := 1 to n
    if i = j then l_ij := 1̄ ⊕ λ(i,j)
             else l_ij := λ(i,j)
for k := 1 to n
    for i,j := 1 to n
        l_ij := l_ij ⊕ (l_ik ⊙ (l_kk)* ⊙ l_kj). (*)
```

The unary iteration operator * is defined as $a^\star := \bigoplus_{i=0}^{\infty} a^i$, with $a^0 := \overline{1}$ and $a^i := a \odot a^{i-1}$.

This algorithm has a time complexity of $O(n^3(T_\oplus + T_\odot) + n^2 T_\star)$, where $T_\oplus$, $T_\odot$ and $T_\star$ are the time complexities for the corresponding operations. Note that for the semiring identified in this paper, these operations do not have a constant time complexity because minimized labels can get arbitrarily large.

Since we want to design a hybrid system consisting of the ATMS and the ASTP, we assume in the following that a unary predicate *nogood* over $\mathcal{P}(\mathcal{A})$ exists with

$$\neg nogood(\emptyset) \wedge$$
$$(nogood(U) \rightarrow (\forall U' : U \subseteq U' \rightarrow nogood(U'))).$$

Let $\mathcal{D}$ be the set of all possible temporal distances. This set makes up a commutative monoid $(\mathcal{D}, +, 0)$ where $\mathcal{D} := R \cup \{-\infty, +\infty\}$ and $+$ is the generalization of real addition on $\mathcal{D}$ according to: $a + (+\infty) = (+\infty) + a = +\infty$, $(-\infty) + (+\infty) = (+\infty) + (-\infty) = +\infty$ and $a + (-\infty) = (-\infty) + a = -\infty$ where $a, b \in R$.

It is now possible to express the set of all temporal distances relative to a set of assumptions as $\mathcal{D}_\mathcal{A} := \mathcal{D} \times \mathcal{P}(\mathcal{A})$ and to order it partially with $\sqsubseteq$:

$$[d_1, U_1] \sqsubseteq [d_2, U_2] \iff d_1 \leq d_2 \wedge U_1 \subseteq U_2$$

The set $\hat{\mathcal{L}}$ of the so called prelabels is

$$\hat{\mathcal{L}} := \{L \mid L \subseteq \mathcal{D}_\mathcal{A} \wedge \exists [d, \emptyset] \in L\}.$$

The idempotent function $m : \hat{\mathcal{L}} \rightarrow \hat{\mathcal{L}}$ minimizes prelabels:

$$m(L) := min_{\sqsubseteq}(L) \setminus \{[d,U] \mid [d,U] \in L \wedge nogood(U)\}.$$

$min_{\sqsubseteq}$ is a function mapping a partial ordered set w.r.t. $\sqsubseteq$ to the set of its minimal elements.

The domain of the semiring, the set $\mathcal{L}$ of the minimized labels, can then be defined as

$$\mathcal{L} := \{m(L) \mid L \in \hat{\mathcal{L}}\}$$

and the semiring $(\mathcal{L}, \oplus, \odot, \overline{0}, \overline{1})$ itself can be specified as follows (with $L_1, L_2 \in \mathcal{L}$):

$$\begin{array}{rcl} L_1 \oplus L_2 & := & m(L_1 \cup L_2) \\ L_1 \odot L_2 & := & m(\{\,[d_1 + d_2, U_1 \cup U_2] \mid \\ & & \qquad [d_1, U_1] \in L_1 \wedge [d_2, U_2] \in L_2\}) \\ \overline{0} & := & \{[+\infty, \emptyset]\} \\ \overline{1} & := & \{[0, \emptyset]\} \end{array}$$

It is easy to show that the requirements for a closed semiring as stated in [AHU74] are fulfilled: $(\mathcal{L}, \odot, \overline{1})$ and $(\mathcal{L}, \oplus, \overline{0})$ are commutative monoides, $\overline{0}$ is absorbing w.r.t. the concatenation operator $\odot$, the summary operator $\oplus$ is idempotent and $\odot$ is distributive over $\oplus$. Summaries over countable infinite sequences exist and $\oplus$ is associative, commutative and idempotent for infinite summaries. Furthermore $\odot$ is distributive over countable infinite summaries.

When applying the iteration operator to a label L_{ii}, it can happen that $[-\infty, U] \in L_{ii}^{\star}$, that is $U \vdash t_i - t_i \leq -\infty$. This constraint is not satisfiable. Therefore the ASTP has no solutions for assumption sets U' where $U' \supseteq U$. In ATMS terminology the assumption set U is a nogood. For our formalization in this case the predicate *nogood* has to be extended by U and all supersets of U:

$$\begin{array}{l} [-\infty, U] \in L_{ii}^{\star} \implies \\ \qquad nogood := nogood \cup \{U' \mid \mathcal{A} \supseteq U' \supseteq U\} \end{array}$$

An ASTP with the constraint set $\mathcal{C}$ can be transformed into a directed graph (called ASTP-graph) weighted with *temporal labels* by creating a node for each variable t_i and creating an edge (t_i, t_j) from t_i to t_j with a temporal label L_{ij} for each pair of nodes: For $i \neq j$:

$$L_{ij} := \{[\infty, \emptyset]\} \oplus \bigoplus \{[d,U] \mid (U \vdash t_j - t_i \leq d) \in \mathcal{C}\}.$$

For $i = j$:

$$L_{ij} := \{[0, \emptyset]\} \oplus \bigoplus \{[d,U] \mid (U \vdash t_i - t_i \leq d) \in \mathcal{C}\}.$$

For each environment $U \subseteq \mathcal{A}$ maximal temporal distances $d(U, t_i, t_j)$ for the time points represented by the variables t_i and t_j can be extracted from the labels L_{ij} of the edges (t_i, t_j) of the graph computed by the Floyd-Warshall algorithm. The corresponding function $d : \mathcal{P}(\mathcal{A}) \times \mathcal{T} \times \mathcal{T} \rightarrow \mathcal{D}$ is defined as:

$$d(U, t_i, t_j) := \min\{d' \mid [U', d'] \in L_{ij} \wedge U' \subseteq U\}.$$

We can use this function to transform temporal labels of edges in the ASTP-graph into logical labels of temporal constraints in the ATMS: U is an environment of the logical label of the implicit temporal constraint

$$t_j - t_i \leq d(U, t_i, t_j),$$

unless it is subsumed by another environment in the ATMS label. It only has to be communicated to the ATMS, if the temporal constraint $t_j - t_i \leq d(U, t_i, t_j)$ is *relevant*. Typically, this is the case if the constraint is mentioned in a justification or queried by the problem solver.

Moreover, two extreme solutions with $t_0 = 0$ can be extracted from the ASTP:

$$\begin{array}{rcl} S_{min}(U) & := & \{(t_i, -d(U, t_i, t_0)) \mid t_i \in \mathcal{T}\} \\ S_{max}(U) & := & \{(t_i, d(U, t_0, t_i)) \mid t_i \in \mathcal{T}\} \end{array}$$

S_{min} (S_{max}) is a solution mapping environments to assignments describing minimal (maximal) distances to t_0.

Minimized temporal labels $L \in \mathcal{L}$ have the following properties:

$$\begin{array}{ll} \forall T \in L, T' \in L : T \sqsubseteq T' \rightarrow T = T' & \text{(minimality)} \\ \forall [d,U] \in L : \neg nogood(U) & \text{(consistency)} \\ \forall U \subseteq \mathcal{A} : \exists [d,U'] \in L : U' \subseteq U & \text{(closedness)} \end{array}$$

The following examples show the effects of the operators (assumption sets like $\{A, B, C\}$ are noted as ABC for brevity).

$$\begin{array}{rcl} m(\{[5, AB], [3, A], [1, C], [\infty, \emptyset]\}) & = & \\ \qquad \{[3, A], [1, C], [\infty, \emptyset]\} & & \\ \{[3, A], [2, B], [\infty, \emptyset]\} \odot \{[1, BC], [5, \emptyset]\} & = & \\ \qquad \{[8, A], [7, B], [3, BC], [\infty, \emptyset]\} & & \\ \{[3, A], [2, B], [\infty, \emptyset]\} \oplus \{[1, BC], [5, \emptyset]\} & = & \\ \qquad \{[3, A], [2, B], [1, BC], [5, \emptyset]\} & & \end{array}$$

We will give an example demonstrating how the Floyd-Warshall algorithm computes the minimized labels once we have shown in the following section how to couple an ATMS and an ASTP.

3 Integration of ATMS and ASTP

The ATMS considers assumption-based temporal constraints of the form $U \vdash t_j - t_i \leq a$ as uninterpreted propositions (nodes) of the form $\lceil t_j - t_i \leq a \rceil$ the logical labels of which include the environment U (explicitly or implicitly, because it was removed by subsumption). Temporal nodes are handled specially; they are *interpreted* by the ASTP.

The interface of the hybrid system to the problem solver corresponds to the interface of the ATMS to the problem solver (cf. [dK86]). In addition, the problem solver can query the AST processor for solutions of the AST problem. Especially, consistent variable assignments can be computed for a given context.

To cope with the special semantics of temporal nodes, we generalize the notion of derivability for the ATMS: A node n holds in an environment U with a set $\mathcal{R}$ of justifications

$$\mathcal{R} \vdash U \rightarrow n$$

if there is a series of node sets $S_1, \ldots, S_m$ with

- $S_1 = tc(U)$,
- $S_{i+1} = tc(S_i \cup \{y \mid \exists (x_1, \ldots, x_k \Rightarrow y) \in \mathcal{R}$ with $x_j \in S_i \ (1 \leq j \leq k)\})$,
- $n \in S_m$.

The function tc denotes the *temporal closure*

$$tc(N) := \bigcup_i \hat{tc}^i(N)$$

of a node set N, where

$$\hat{tc}(N) := N \cup \{\lceil t_j - t_i \leq d \rceil \mid \lceil t_k - t_i \leq a_{ik} \rceil \in N \wedge \lceil t_j - t_k \leq a_{kj} \rceil \in N \wedge d \geq a_{ik} + a_{kj}\}.$$

Given the formal specification of the integration of the ATMS and the ASTP, how can the two systems be combined in practice? What would an implementation of it look like? It is evident that changes in the logical labels of temporal nodes that are computed by the ATMS have to be collected and communicated to the ASTP. This now causes changes in the temporal labels of edges in the ASTP-graph. Hence, the ASTP-graph must be recompleted with the Floyd-Warshall algorithm. This causes another change in the temporal label of an ATMS-node, if the newly discovered constraint is relevant, and the game starts again.

An efficient implementation of the hybrid system (in the sequel called ATTMS — the assumption-based temporal truth maintenance system) requires changes to the ATMS label propagation algorithm and the generalized Floyd-Warshall algorithm.

The ATMS works incrementally: one justification at a time is added to the dependency network. After each addition a label update is invoked, even when the problem solver is not currently interested in the label of a node.

The generalized Floyd-Warshall algorithm as realized in [AHU74] does not work incrementally. Using it as is for the ASTP therefore means computing all temporal labels after every change which is not acceptable.

As will be seen in a moment, both modules of the hybrid system have to be adapted. This adaption essentially affects the following areas:

Processing of nogoods: The ATMS and the ASTP must maintain a common set of nogoods. This set of nogoods is taken into consideration in the algorithm used for solving an ASTP (cf. the definition of the minimizing function m and its use in the definition for the operators $\oplus$ and $\odot$).

The ASTP can discover new nogoods. As shown previously, this happens when the iteration operator computes a temporal label with the temporal distance $-\infty$. In this situation all the newly discovered nogoods are automatically communicated to the ATMS.

Computing the temporal labels in the ASTP-graph: Computing the temporal labels from scratch can be very inefficient. In general, it is more efficient to use an incremental algorithm for the update of the ASTP labels. There are two ways to develop an incremental label update algorithm.

First, the generalized Floyd-Warshall algorithm can be made more "incremental". In [RR91] it is shown that it is not possible to construct a real δ-incremental algorithm (having a time bound depending only on the size of the incremental change of the input and the minimal change of the output) for the general all pairs shortest path problem and thus for the path problem on closed semirings. But it is very easy to develop an algorithm with a time complexity of $O(n^2(T_\oplus + T_\odot + T_\star))$ for an edge insertion operation using the approach described in [IK83].

During label propagation in the ATMS, usually more than one label of a temporal constraint is updated. All these updates are collected and added to the ASTP at the same time. The outer loop of the Floyd-Warshall algorithm only has to process the endpoints of added edges.

Another big efficiency improvement can be obtained by incrementalizing the semiring operations, analogous to the incremental ATMS label update algorithm. A formal description of this incremental version is left out for brevity and can be found in [Gei94].

Processing the information resulting from temporal propagation efficiently: Even an "incremental" Floyd-Warshall algorithm as described above works in a kind of batch processing mode. With each batch a number of temporal labels is updated, which can be transformed into logical labels of temporal constraints as previously shown. The ATMS therefore must update more than one label at a time. Technically these updates can be seen as new justifications for the ATMS.

The ATMS is not interested in all of the temporal constraints maintained by the ASTP, but only in temporal constraints used in justifications or queries of the problem solver. Only changes of labels belonging to these constraints must be communicated to the ATMS.

Nevertheless, in general, a set of justifications has to be communicated to the ATMS. We call the problem of efficiently adding a set of justifications to the dependency network as the *bulk update problem*. A solution for the bulk update problem for the ATMS is shown in [BG94, Gei94]. The two central ideas for this algorithm are avoiding redundant label products

and an optimal serialization of the label updates in the dependency network.

Typically, there are a lot of common antecedents in large justification sets as occurring in the justification sets communicated by the ASTP. Computing label products of these antecedents as needed during the ATMS label propagation algorithm leads to a waste of computation when done several times for the same antecedents. With a fast heuristic common subexpression elimination algorithm these redundant computations can mostly be recognized and avoided.

In order to do that, label propagation has to be synchronized so that the set of pending updates on which the common subexpression elimination is performed is as large as possible. A side effect of the synchronization is the minimization of the number of propagation waves started at each node. This can be achieved by using the partial order underlying the dependency network. A non-trivial subproblem is the incremental actualization of this topological information based on the structure of the strongly connected components of the dependency network. An efficient δ-incremental algorithm maintaining this information as new justifications are added was developed.

4 An example

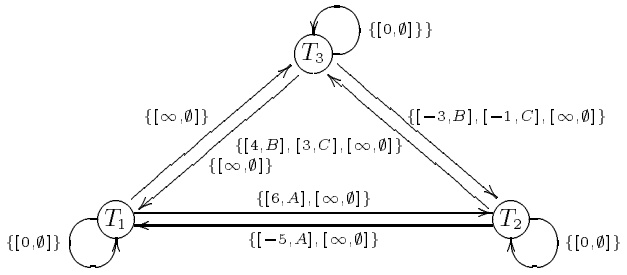
Suppose the following justifications are communicated to the ATTMS (assumptions are denoted with capital letters)[3]:

$(A \wedge B \Rightarrow r)$, $(B \wedge C \Rightarrow \bot)$, $(A \Rightarrow \lceil 5 \leq T_2 - T_1 \leq 6 \rceil)$,
$(B \Rightarrow \lceil 3 \leq T_3 - T_2 \leq 4 \rceil)$, $(C \Rightarrow \lceil 1 \leq T_3 - T_2 \leq 3 \rceil)$,
$(D \wedge \lceil T_1 - T_3 \leq -6 \rceil \Rightarrow s)$

Now the label propagation in the ATMS is started. After it has is finished, the logical labels of the nodes in the ATMS are updated to[4]

$\langle r, \{AB\} \rangle$, $\langle \lceil T_1 - T_3 \leq -6 \rceil, \{\} \rangle$, $\langle s, \{\} \rangle$,
$\langle \lceil T_2 - T_3 \leq -3 \rceil, \{B\} \rangle$, $\langle \lceil T_2 - T_3 \leq -1 \rceil, \{B, C\} \rangle$.

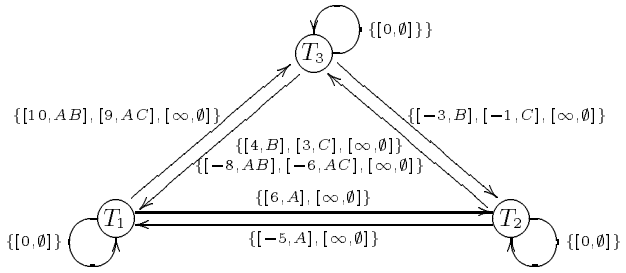
As this causes changes in the logical labels of some temporal nodes, the corresponding temporal labels in the ASTP-graph change as well. The ASTP-graph then looks as follows:



[3] $(A \Rightarrow \lceil 5 \leq T_2 - T_1 \leq 6 \rceil)$ is an abbreviation for the three justifications
$(A \Rightarrow \lceil T_2 - T_1 \leq 6 \rceil)$, $(A \Rightarrow \lceil T_1 - T_2 \leq -5 \rceil)$,
$(\lceil T_2 - T_1 \leq 6 \rceil \wedge \lceil T_1 - T_2 \leq -5 \rceil \Rightarrow \lceil 5 \leq T_2 - T_1 \leq 6 \rceil)$.

[4] A node n with label $l(n)$ is written as $\langle n, l(n) \rangle$

Now the Floyd-Warshall algorithm must recomplete the temporal labels in the ASTP-graph:



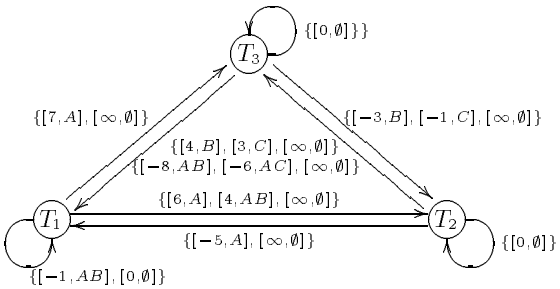
This causes several logical labels to change. Here are the labels of a few selected nodes in the ATMS:

$\langle r, \{AB\} \rangle$, $\langle \lceil T_1 - T_3 \leq -6 \rceil, \{AB, AC\} \rangle$,
$\langle s, \{ABD, ACD\} \rangle$, $\langle \lceil T_3 - T_1 \leq 9 \rceil, \{AC\} \rangle$.

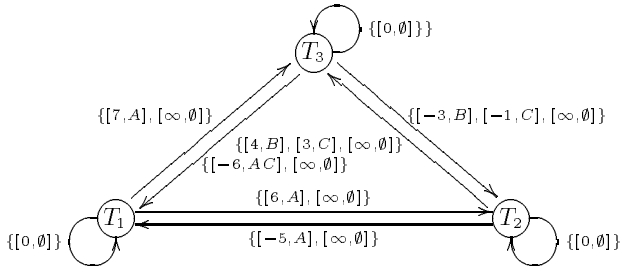
Suppose that a new justification $(A \Rightarrow \lceil T_3 - T_1 \leq 7 \rceil)$ is added to the ATTMS now. As $[7, A] \sqsubseteq [10, AB]$ and $[7, A] \sqsubseteq [9, AC]$, the temporal label of the ASTP-edge $T_1 \rightarrow T_3$ can be simplified to

$$\{[7, A], [\infty, \emptyset]\}.$$

After the subsequent completion of the temporal labels with the Floyd-Warshall algorithm, the ASTP-graph looks at follows:



Since the label of the ASTP-edge $T_1 \rightarrow T_1$ is containing a relative temporal distance $[-1, AB]$ with a negative distance value, the environment AB is discovered as being inconsistent. This produces a new nogood AB for the ATTMS and all environments that are a superset of this nogood have to be removed from the logical labels in the ATMS. In addition, all relative distances with environments which are a superset of the new nogood have to be removed from the temporal labels in the ASTP. Eventually, the ASTP-graph looks as follows:



The logical labels of the interesting nodes in the ATMS are then given by:

$$\langle r, \{\}\rangle,\ \langle [T_1 - T_3 \leq -6], \{AC\}\rangle,\ \langle s, \{ACD\}\rangle.$$

5 Related work

Only a few approaches can be found in the literature which attempt to merge assumption-based reasoning with temporal reasoning. All these systems share the application domain "model-based diagnosis". Propositions are associated with time points or time intervals in order to model states at certain time points or during time intervals.

A first attempt to integrate temporal reasoning into an ATMS was made in [DF89]. In this approach, propositions, assumption sets and intervals are related via a global table. Symbolically represented time intervals are propagated similar to assumptions along time-independent logical justifications resulting in expressions over intervals. The temporal dimension is just a second indexing scheme besides the assumption sets.

HEART [JR90] combines propositions and time intervals with start and end points instantiated with numbers. The result of this combination (called an *episode*) is treated as a node. Assumptions as well as antecedents and the conclusion of a justification are episodes. "Environments" are sets of episodes. As the assumptions are structured in a purely propositional and an interval component, the notion of label minimality is generalized. In order to keep "labels" minimal, episodes with the same propositions and overlapping intervals are merged. Redundant episodes (based on interval inclusion) are removed.

In an eATMS justification [TL93] every proposition P is associated with a time point variable t or the set of all time points α. Justifications have the form

$$P_1{:}t_1 \wedge \ldots \wedge P_k{:}t_k \wedge R(t_1, \ldots, t_k) \Rightarrow Q{:}f(t_1, \ldots, t_k),$$

where the t_i are time point variables or α, R is a predicate over tuples of time points enabling the specification of temporal relations and f is a function mapping tuples of time points to time points. Assumptions are time-independent. The label of a time-independent proposition is a set of ordinary ATMS labels indexed with time points instantiated with numbers and are only minimized w.r.t. the α entry. The ATMS propagation algorithm is extended to handle justifications of the form described above. One disadvantage of this approach is that the termination of the propagation algorithm is not guaranteed because there are temporal variables in justifications allowing infinite deductions. Another disadvantage is the restriction to time points.

In the TARMS [HNP91], the most complex of these hybrid systems, the assumptions are time-independent as well. The temporal information is represented in the labels of time-independent propositions, which are sets of ordinary ATMS labels indexed with fixed time intervals. These labels are minimized by merging adjacent intervals when they are associated to the same ATMS labels. The justifications are splitted in two parts: an ATMS-like justification and a temporal method. The temporal method determines the intervals used in the consequents' label depending on the labels of the antecedents. Using different temporal methods, a variety of temporal relationships can be modelled.

None of the systems mentioned here is able to process temporal constraints as usual temporal reasoning systems do because they are not able to solve temporal constraint satisfaction problems. Also, inconsistencies in the temporal dimension are not handled as nogoods in the ATMS. Furthermore these systems are not suitable for scheduling applications because no metric temporal information can be processed.

Time map management systems as the well known TMM [DM87] do not support the simultaneous management of multiple contexts. They are therefore not well-suited as support systems that have to manage assumptions explicitly which is important in application domains like planning and scheduling.

A main difference between our approach and the approaches cited is the treatment of temporal constraints as nodes, which can be justified in the same manner as other nodes and therefore can have different status in different contexts.

6 Summary and future work

We have presented a hybrid system integrating assumption-based logical and temporal reasoning. This system is a common generalization of the ATMS and an assumption-based temporal reasoning system. The resulting hybrid system (the ATTMS) is especially suited to support integrated planning and scheduling.

The system is implemented already in Common Lisp, but some more work has to be done in order to evaluate and improve the performance of the overall system. We are currently investigating the formal complexity of subalgorithms and of the whole system.

We are also thinking about what an interface to a problem solver on a higher level of abstraction might look like. Other temporal reasoning approaches including Allen's interval algebra [All83] will have to be examined whether they can be extended in an assumption-based manner and integrated into the ATMS. The approach of Allen in a sense also uses the Floyd-Warshall algorithm, which is not complete for Allens interval algebra. It would be interesting to investigate the restriction to the maximal subclass ORD-Horn of the interval algebra found by Nebel [NB93] for which the Floyd-Warshall algorithm is complete.

Last but not least we are evaluating the practical suitability of our approach based on several larger planning and scheduling problems.

References

[AHU74] Aho, A. V.; Hopcroft, J. E.; Ullman, J. D.: *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, MA, 1974.

[All83] Allen, J. F.: *Maintaining Knowledge about Temporal Intervals. Communications of the ACM*, 11(26):832–843, 1983.

[BG94] Beckstein, C.; Geisler, T.: *An Improved Dependency Network Management Algorithm for the ATMS.* Technical report, IMMD8, Universität Erlangen-Nürnberg, 1994. In preparation.

[BLS92] Beetz, M.; Lindner, M.; Schneeberger, J.: *Temporal Projection for Hierarchical, Partial-order Planning.* Technical Report AIDA-92-15, FG Intellektik, TH Darmstadt, 1992.

[CLR90] Cormen, T. H.; Leiserson, C. E.; Rivest, R. D.: *Introduction to Algorithms.* MIT Press/McGraw-Hill, Cambridge, MA, 1990.

[DF89] Dressler, O.; Freitag, H.: *Propagation of Temporally Indexed Values in Multiple Contexts.* In: *Proc. GWAI-89, Eringerfeld*, pages 88–99, 1989.

[dK86] de Kleer, J.: *1. An Assumption-based TMS, 2. Extending the ATMS, 3. Problem solving with the ATMS. Artificial Intelligence*, 28:127–224, 1986.

[DM87] Dean, T. L.; McDermott, D. V.: *Temporal Data Base Management. Artificial Intelligence*, 32:1–55, 1987.

[DMP91] Dechter, R.; Meiri, I.; Pearl, J.: *Temporal Constraint Networks. Artificial Intelligence*, 49:61–95, 1991.

[Gei94] Geisler, T.: *Ein anwendungsunabhängiges Unterstützungssystem zum integrierten annahmenbasierten temporalen Schließen.* Diplomarbeit, Universität Erlangen-Nürnberg, IMMD8, 1994.

[HNP91] Holtzblatt, L. J.; Neiberg, M. J.; Piazza, R. L.: *Temporal Reasoning in an Assumption Based Reason Maintenance System.* Technical Report M91-22, MITRE, Bedford, MA, 1991.

[IK83] Ibaraki, T.; Katoh, N.: *On-Line Computation of Transitive Closures of Graphs. Information Processing Letters*, 16:95–97, 1983.

[JR90] Joubel, C.; Raiman, O.: *How Time Changes Assumptions.* In: *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI '90), Stockholm*, 1990.

[NB93] Nebel, B.; Bürckert, H.-J.: *Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra.* Technical Report RR-93-11, DFKI, Saarbrücken, 1993.

[RR91] Ramalingan, G.; Reps, T.: *On the Computational Complexity of Incremental Algorithms.* Technical Report TR 1033, University of Wisconsin, Madison, 1991.

[TL93] Tatar, M. M.; Letia, I. A.: *Embedding Temporal Reasoning into the ATMS Framework.* In Puppe, F.; Günther, A., editors: *Expertensysteme '93 (XPS-93), Hamburg*, pages 276–282, 1993.