Temporal Reasoning with Constraints on Fluents and Events *

Eddie Schwalb, Kalev Kask, Rina Dechter

Department of Information and Computer Science
University of California, Irvine CA 92717
{eschwalb,kkask,dechter}@ics.uci.edu

Abstract

We propose a propositional language for temporal reasoning that is computationally effective yet expressive enough to describe information about fluents, events and temporal constraints. Although the complete inference algorithm is exponential, we characterize a tractable core with limited expressibility and inferential power. Our results render a variety of constraint propagation techniques applicable for reasoning with constraints on fluents.

1 Introduction

Consider the issues raised by the following "story". At 8:00 the microfilm was deposited in the safe and at 11:00 the microfilm was gone. John was at the bar between 8:10 - 8:30 and between 9:10 -12:00. He was also at the poker table between 8:35 - 9:00. Fred was at the bar between 8:30 - 10:00 and between 10:45 - 12:00. The bar opened at 7:30 and closed at 12:00. We know that at least 15 minutes are required to take the microfilm and return to the bar.

Given the story above, we are interested in answering queries such as "Does the story entail that Fred took the microfilm ?" and "What are *all* the possible scenarios in this story ?". We wish to capture the human ability to answer such queries without information about speed of movement or distances.

We wish to describe information conditioned on occurrences of events. For instance, a sentence like "at least 15 minutes are required to take the microfilm and return to the bar" should be accounted for only if someone took the film; a query like "When did John take the microfilm ?" assumes that John took the microfilm.

*This work was partially supported by NSF grant IRI-9157636, by Air Force Office of Scientific Research grant AFOSR 900136 and by grants from TOSHIBA of America and Xerox

In classical propositional logic the atomic entities are propositions. In dynamic environments the atomic entities are called *fluents*. They may repeatedly change their value as events occur and are functions of the situation or time. Our work builds upon temporal languages proposed by McDermott and Dean [11, 5], Allen [2], Kowalski and Sergot [9] and Shoham [14]. We accommodate various constructs proposed in these languages (time points and intervals). Our main goal, however, is to equip a temporal language with a computationally manageable inference engine.

The primary task of any reasoning system is determining consistency of the given theory. For temporal languages this means determining the consistency of sentences involving a combination of temporal and propositional constraints. For example, consider the statement "Either Bob or Mary must tend the bar, but Bob has to leave on an errand and Mary has an appointment with the doctor". How do we infer that Bob's errand must be either before or after Mary's appointment ?

In this paper we propose a temporal language whose inference engine is based on qualitative and quantitative temporal constraints [1, 6, 7, 12, 15, 16]. Decoupling the propositional and temporal constraints provides us with inference algorithms that are based on propositional satisfiability and temporal constraint satisfaction, and allows us to identify a useful tractable core.

The proposed language, called `HOT`, is defined over $Holds$, $Occurs$ and $Temporal$ propositions. We assume that events are instantaneous and serve as the only time points at which fluents may change their values. We use $Holds(F, E_1, E_2)$ to state that the fluent F is $true$ between events E_1 and E_2, $Occurs(E)$ to state that event E occurred, and $Time(E)$ for the time of its occurrence. We focus on the tasks of deciding consistency and computing consistent scenarios, which enables us to answer entailment and other queries of interest (such as inferring whether certain events *must*, *might*, or *could not* have occurred). Our language has a tractable core which allows us to make weak (sound but incomplete) inferences. When augmented with additional axioms, our language yields sound and complete inference algorithms at the expense of increased computational complexity.

The paper is organized as follows. Section 2 describes the syntax and semantics of our language. Section 3 gives an alternative propositional semantics for the language. Section 4 introduces a new model of conditional temporal constraint networks that serves as the computational engine.

2 The Language

Our language, called `HOT`, is a set of sentences over *Holds*, *Occurs* and *Temporal* propositions. We use two (disjoint) sets of symbols, *fluent symbols* and *event symbols*. A fluent is a propositional function of time and a *fluent literal* is either a fluent symbol or its negation. An event E is a pair $(Occurs(E), Time(E))$, where $Occurs(E)$ is a propositional variable that is assigned *true* iff E occurred, and $Time(E)$ is a real valued variable that specifies the time E occurred. We use two special events: E_{begin}, for "the beginning of the world", and E_{end}, for "the end of the world". Unless otherwise noted, we use right-sided half open intervals $[a, b)$ for reasons to be clarified later.

In addition to *Occurs* propositions there are *Holds* and *Temporal* propositions.

A *Holds* proposition has the form

$$Holds(\varphi,\ E_i, E_j) \tag{1}$$

where E_i and E_j are event symbols and $\varphi = F_1 \vee \cdots \vee F_k$ is a disjunction of fluent literals. We also use the following abbreviations: If $E_i = E_j$ we write (1) as $Holds(\varphi,\ E_i)$. If $E_i = E_{begin}$ we write (1) as $Holds(\varphi,$ **before** $E_j)$. If $E_j = E_{end}$ we write (1) as $Holds(\varphi,$ **after** $E_i)$. If $E_i = E_j = E_{begin}$ or $E_i = E_j = E_{end}$ or both $E_i = E_{begin}$ and $E_j = E_{end}$ we write (1) as **initially** φ, **eventually** φ, **always** φ respectively.

A *Temporal* proposition is a qualitative or quantitative temporal constraint over time points and time intervals [12]. Given a pair of events E_i, E_j, I_{E_i,E_j} denotes an half-open interval that begins with (includes) $Time(E_i)$ and ends with (excludes) $Time(E_j)$. A temporal proposition is a constraint having one of three forms:

1. A point-point constraint

$$Time(E_j) - Time(E_i)\ \in\ I_1 \cup I_2 \cup \cdots \cup I_k \tag{2}$$

where $I_1, \ldots, I_k$ are intervals (over real numbers), specified by their end points. We also use the shortcuts $Time(E_j)\ \in\ I_1 \cup I_2 \cup \cdots \cup I_k$, $Time(E_j) = t$ and $Time(E_j) \geq Time(E_i)$ to have the obvious meaning.

2. A point-interval constraint between the time point at which event E_i occurred and the interval that begins with E_j and ends with E_k,

$$Time(E_i)\ \{R_1, \ldots, R_m\}\ I_{E_j,E_k} \tag{3}$$

where $R_1, \ldots, R_m \in$ { **before, starts, during, finishes, after** }.

3. An interval-interval constraint

$$I_{E_i,E_j}\ \{R_1, \ldots, R_m\}\ I_{E_p,E_q} \tag{4}$$

where $R_1, \ldots, R_m,\ m \leq 13$ are distinct and

$$R_1, \ldots, R_m \in \left\{ \begin{array}{c} \textbf{before, after, meets, met-by,} \\ \textbf{overlaps, overlapped-by,} \\ \textbf{during, contains, equals,} \\ \textbf{starts, started-by,} \\ \textbf{finishes, finished-by} \end{array} \right\}.$$

Sentences in `HOT` are conjunctive normal form (CNF) formulas over *Holds*, *Occurs* and *Temporal* propositions as their atoms.

Example 1: Consider the story in the introduction. The sentence "At 8:00 the film was deposited in the safe and at 11:00 the film was gone" is described by

$$\begin{array}{l} Holds(Film\_in\_safe,\ Film\_deposited) \wedge \\ (Time(Film\_deposited) = 8:00) \wedge \\ Holds(\neg Film\_in\_safe,\ Film\_checked) \wedge \\ (Time(Film\_checked) = 11:00). \end{array}$$

The sentence "at least 15 minutes are required to take the microfilm and return to the bar" is described by

$$\begin{array}{l} Occurs(John\_take\_film) \rightarrow (Time(end\_John\_go\_safe) \\ -Time(begin\_John\_go\_safe) \in\ [15, \infty]) \end{array}$$

A similar sentence can be described for Fred.

2.1 Semantics

An interpretation of a formula in `HOT` is a quadruple $< \mathcal{F}, M_f,\ \mathcal{E}, M_e >$, where $\mathcal{F}$ is a set of two-valued functions of time; M_f is a mapping $M_f\ :\ F \mapsto \mathcal{F}$ of fluent symbols into functions in $\mathcal{F}$; $\mathcal{E}$ is a subset of event symbols and M_e is a mapping $M_e\ :\ \mathcal{E} \mapsto \Re$ of events in $\mathcal{E}$ into real valued time points. The value of $M_f(\varphi)$ may change only when events occur, namely at a time point $t = M_e(E)$ for some event E.

Intuitively, $\mathcal{F}$ is a set of fluents that corresponds to fluent symbols used in the formula and $\mathcal{E}$ is the set of events that actually occurred, mapped to the time points at which each of them occurred.

Definition 1: An interpretation is a *scenario* (or a model) of a formula if all its clauses are assigned the truth value *true* under the following rules of evaluation:

1. $Occurs(E)$ is *true* iff $E \in \mathcal{E}$.
2. We extend M_f to disjunction and negation, $M_f(F_1 \vee \cdots \vee F_k) = M_f(F_1) \vee \cdots \vee M_f(F_k)$, $M(\neg F) = \neg M(F)$.

3. A holds proposition $Holds(\varphi,\ E_i, E_j)$ is $true$ iff $E_i, E_j \in \mathcal{E}$, and (a) in case $E_i = E_j$ then $M_f(\varphi)(M_e(E))$ is $true$, (b) in case $E_i \neq E_j$ then $M_e(E_i) < M_e(E_j)$ and for any t such that $M_e(E_i) \leq t < M_e(E_j)$, $M_f(\varphi)(t)$ is $true$.

4. A temporal proposition is $true$ iff the events specified occurred and the temporal constraint is satisfied, namely
   - (a) a *point-point* temporal proposition (2) is $true$ iff $E_i, E_j \in \mathcal{E}$, and $M_e(E_j) - M_e(E_i) \in I_1 \cup I_2 \cup \cdots \cup I_k$.
   - (b) a *point-interval* temporal proposition (3) is $true$ iff $E_i, E_j, E_k \in \mathcal{E}$ and $M_e(E_j) < M_e(E_k)$ and one of the relations $R_1, \ldots, R_m$ holds.
   - (c) an *interval-interval* temporal proposition (4) is $true$ iff $E_i, E_j, E_p, E_q \in \mathcal{E}$ and $M_e(E_i) < M_e(E_j)$ and $M_e(E_p) < M_e(E_q)$ and one of the relations $R_1, \ldots, R_m$ holds.

5. The truth value of the clauses and the CNF formula is evaluated with respect to the truth values of occurs, holds and temporal propositions using standard rules of evaluation.

A formula is *s-satisfiable* iff it has a scenario.

Note that $Holds(\varphi_1 \wedge \varphi_2, E_i, E_j) \equiv Holds(\varphi_1, E_i, E_j) \wedge Holds(\varphi_2, E_i, E_j)$. However, $Holds(\varphi_1 \vee \varphi_2, E_i, E_j)$ is obviously not equivalent to $Holds(\varphi_1, E_i, E_j) \vee Holds(\varphi_2, E_i, E_j)$. Also note that $Holds(\neg\varphi, E_i, E_j)$ is not equivalent to $\neg Holds(\varphi, E_i, E_j)$.

The formula $Holds(F, E_i, E_j) \wedge Holds(\neg F, E_i)$ is inconsistent but $Holds(F, E_i, E_j) \wedge Holds(\neg F, E_j)$ is consistent because the interval I_{E_i,E_j} is half-open. If we used closed intervals, $Holds(F, E_i, E_j) \wedge Holds(\neg F, E_j, E_k)$ would have been inconsistent and the values of the fluents would not be allowed to change when events occur. If we had used open intervals, specifying $Holds(F, E_i)$ would have been useless since it does not induce a constraint on the value of $M_f(F)(t)$ for the open interval $t \in (Time(E_i), Time(E_j))$.

An occurs, holds, or temporal proposition q is entailed by Ψ, denoted $\Psi \models q$, iff it is true in all scenarios of Ψ. As usual, $\Psi \models q$ iff $\Psi \wedge \neg q$ is inconsistent.

Example 2 : Consider the statement "John was at the bar from 8:10 to 8:30". It is described by the formula $\Psi = Holds(John\_at\_bar, 8{:}10, 8{:}30)$.[1] $\Psi \models Holds(John\_at\_bar, 8{:}15, 8{:}25)$ but $\Psi \not\models Holds(John\_at\_bar, 8{:}15, 8{:}35)$ because the value of the fluent $John\_at\_bar$ is not constrained after 8:30 and thus it can be either true or false.

[1] For the sake of convenience we will use real time points as events.

For the rest of this paper we will restrict our treatment to HOT sentences whose $Holds(\varphi, E_i, E_j)$ propositions we call *simple*, namely holds propositions in which φ is a single fluent literal unless $E_i = E_{begin}$ and $E_j = E_{end}$ (i.e. $Holds(\varphi, \mathbf{always})$). General holds propositions introduce computational complications which we will not address in this paper.

3 Propositional Semantics for HOT

In this section we address the task of deciding whether a formula is s-satisfiable. We wish to show that the task of deciding s-satisfiability and finding a scenario reduces to a two-step process of propositional satisfiability and temporal constraint satisfaction. Although both of these tasks are NP-complete, such a reduction opens the way for using known heuristics and known tractable classes. The idea is to view Ψ as a propositional CNF formula. Once a propositional model is available, using temporal constraint satisfaction we can determine whether all temporal constraints, specified by temporal propositions assigned $true$ by the model, can be satisfied simultaneously.

Definition 2: [p-model] Given a set of event and fluent symbols, a *p-interpretation* is a truth value assignment to holds, occurs and temporal propositions when viewed as propositional variables. A p-interpretation is a p-model of a HOT formula Ψ iff it is a propositional model of Ψ and the set of temporal constraints specified by the temporal propositions assigned $true$ is consistent.

Upon trying this approach we see immediately that this process may yield p-models that do not correspond to any real scenario. The reason is twofold: holds propositions impose implicit temporal constraints that are not explicit in the formula, and temporal propositions should be assigned $true$ iff the temporal constraint they induce is satisfied (p-models capture only one-way implication). For instance, a formula consisting of just one holds proposition $Holds(F, E_i, E_j)$ will have a p-model that allows any assignment to temporal variables $Time(E_i)$ and $Time(E_j)$, since there is no explicit temporal proposition specifying $Time(E_i) < Time(E_j)$. In order to avoid these superfluous p-models, we augment Ψ with axioms that explicate the intended meaning.

In the following paragraphs we will augment a formula Ψ with additional HOT sentences that will be called *axioms*. The resulting augmented theory Ψ' describes the same set of scenarios as the original theory Ψ. However, every p-model of Ψ' corresponds to a set of scenarios of Ψ and every scenario of Ψ corresponds to a p-model of Ψ'.

Definition 3: [axiom set A_1] Given a formula Ψ, the axiom set A_1 has four parts:

1. For all events add $Time(E_{begin}) < Time(E) < Time(E_{end})$, and for all pairs of events E_i, E_j add $I_{E_i,E_j}\{equals\}I_{E_j,E_i}$.
2. For every event E specified in a temporal proposition T we add the axiom $T \rightarrow Occurs(E)$.
3. For every holds proposition $Holds(\varphi, E_i, E_j)$ of Ψ we include sentences stating that if a holds proposition is true, the corresponding events should have occurred and in the intended order:

$$Holds(\varphi, E_i, E_j) \rightarrow Occurs(E_i) \wedge Occurs(E_j),$$
$$Holds(\varphi, E_i, E_j) \rightarrow (Time(E_i)\{starts\}I_{E_i,E_j}),$$
$$Holds(\varphi, E_i, E_j) \rightarrow (Time(E_j)\{finishes\}I_{E_i,E_j}).$$

4. For every pair of holds propositions with opposing fluents, $Holds(F, E_i, E_j)$ and $Holds(\neg F, E_p, E_q)$, we add a sentence stating that the two intervals are disjoint. In general we will add

$$Holds(F,\ E_i, E_j) \wedge Holds(\neg F,\ E_p, E_q) \rightarrow (I_{E_i,E_j}\{before, meets, met\text{-}by, after\}I_{E_p,E_q}).$$

although there are special cases that are simpler.

The next set of axioms deals with the complications introduced by disjunctive holds propositions. Here is an example.

Example 3: Consider the example statement "Either Bob or Mary must tend the bar, but Bob has to leave on an errand and Mary has an appointment with the doctor". It can be represented by the formula $\Psi =$

$$\begin{array}{c} Holds(Bob\_tend\_bar \vee Mary\_tend\_bar,\ \mathbf{always}) \wedge \\ Holds(\neg Bob\_tend\_bar, begin\_errand, end\_errand) \wedge \\ Holds(\neg Mary\_tend\_bar, begin\_apnt, end\_apnt) \end{array} \quad (5)$$

In order to guarantee consistency of (5), it is necessary that the intervals of Bob's errand and Mary's appointment be disjoint. Otherwise, there will be a time point at which both Bob_tend_bar and $Mary\_tend\_bar$ are $false$, contradicting

$$Holds(Bob\_tend\_bar \vee Mary\_tend\_bar,\ \mathbf{always}).$$

The set of axioms A_2, defined next, includes a constraint that enforces those intervals to be disjoint.

Definition 4: [axiom set A_2] Given a formula Ψ, if there exists a holds proposition $h_0 = Holds(F_1 \vee \ldots \vee F_k,\ \mathbf{always})$ then for any set of k holds propositions $\{h_i = Holds(\neg F_i,\ E_{p_i}, E_{q_i}) \mid 1 \leq i \leq k\}$ we include the sentence

$$h_0 \wedge h_1 \wedge \ldots \wedge h_k \rightarrow \bigvee_{i<j\leq k}(I_{E_{p_i},E_{q_i}}\{before, meets, met\text{-}by, after\}I_{E_{p_j},E_{q_j}})$$

although there are special cases that are simpler.

Example 4: To illustrate the utility of axioms A_1 and A_2, consider the statement "John or Fred was always at the bar, but Fred was not at the bar after 8 : 00 and John was not at the bar at 10 : 00" which can be represented by the formula

$$\begin{array}{l} Holds(John\_at\_bar \vee Fred\_at\_bar,\ \mathbf{always}) \wedge \\ Holds(\neg Fred\_at\_bar,\ \mathbf{after}\ 8{:}00) \wedge \\ Holds(\neg John\_at\_bar,\ 10{:}00) \end{array}$$

The axiom set A_1 includes $Time(E_{begin}) < 8{:}00 < 10{:}00 < Time(E_{end})$, and the axiom set A_2 includes

$$\begin{array}{l} Holds(John\_at\_bar \vee Fred\_at\_bar,\ E_{begin}, E_{end}) \wedge \\ Holds(\neg Fred\_at\_bar,\ 8{:}00, E_{end}) \wedge \\ Holds(\neg John\_at\_bar,\ 10{:}00) \rightarrow \\ \quad (I_{8:00,E_{end}}\{before, meets, met\text{-}by, after\}10{:}00). \end{array}$$

Since, the constraints introduced by A_1 and A_2 are inconsistent, the statement is inconsistent.

The set of axioms A_1 and A_2 is not sufficient to eliminate all superfluous p-models. They do guarantee that for every holds proposition $Holds(F, E_i, E_j)$ assigned $true$ we can assign $M_f(F)(t) = true$ for every $t \in I_{E_i,E_j}$ without creating conflicts. However, A_1 and A_2 do not guarantee that when the holds proposition is assigned $false$, the negation of the intended constraint is satisfied in every p-model. To guarantee completeness we provide yet another set of axioms, denoted A_3, which specify that the truth value of a holds proposition is inherited by sub- and super-intervals and that a temporal proposition is assigned $true$ iff the temporal constraint is satisfied.

Definition 5: [axiom set A_3] Given a formula Ψ, for every disjunction of fluent symbols $\varphi = F_1 \vee ... \vee F_k$ used in the formula, all fluent symbols F, every set of events E_i, E_j, E_p, E_q and every temporal proposition T we include:

$$\neg Holds(\psi, E_i, E_j) \wedge Occurs(E_i) \wedge Occurs(E_j) \rightarrow \bigvee_{\forall p,q}\{(I_{E_p,E_q} \subseteq I_{E_i,E_j}) \wedge\ Holds(\neg\ \psi, E_p, E_q)\} \quad (6)$$

$$Holds(\psi, E_i, E_j) \wedge (I_{E_p,E_q} \subseteq I_{E_i,E_j}) \rightarrow Holds(\psi, E_p, E_q) \quad (7)$$

[2] $$\neg T \bigwedge_{\forall E \text{ specified in } T} Occurs(E) \rightarrow \overline{T} \quad (8)$$

where $\psi \in \{\varphi, F\}$, $\subseteq$ stands for the constraint {starts, during, finishes, equals} and $\overline{T}$ denotes the complement of the temporal constraint T.

For example, if $T = (I_1\{\text{starts,during,finishes}\}I_2)$ then $\overline{T} = (I_1$ {before, after, meets, met-by, overlaps, overlapped-by, contains,finished-by,started-by,equals}$I_2)$.

Lemma 1: *The size of axiom sets A_1, A_2 and A_3 is at most $O(n^2)$, $O(nk^2(\frac{n}{k})^k)$ and $O(n \cdot n_e^4)$ respectively, where n is the size of the theory axioms are added to, n_e is the number of event symbols and k is the maximum number of fluent literals in a disjunctive holds proposition.*

---

[2] This axiom enables contrapositive reasoning with temporal propositions.

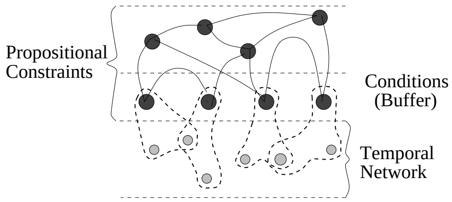

Figure 1: The structure of CTNs.

We use $\Psi \cup A_1 \cup A_2 \cup A_3$ to denote the closure under axioms A_1, A_2 and A_3.

Lemma 2: *The* HOT *formulas* Ψ *and* $\Psi' = \Psi \cup A_1 \cup A_2 \cup A_3$ *are equivalent with respect to s-satisfiability.*

Theorem 1: *Every p-model of* $\Psi \cup A_1 \cup A_2 \cup A_3$ *corresponds to a set of scenarios of* Ψ *and every scenario of* Ψ *corresponds to a p-model of* $\Psi \cup A_1 \cup A_2 \cup A_3$.

4 Conditional Temporal Networks

The notion of p-models decouples the propositional constraints from the temporal constraints and enables us to discuss them in isolation. We call this framework *conditional temporal networks.*

Definition 6: A *conditional temporal network* (CTN) has two types of variables: propositional and temporal (point and interval), and two types of constraints: propositional and temporal. Every temporal constraint T is associated with a unique propositional variable C, called its condition. A *conditional temporal constraint* is a pair $(C : T)$. A propositional constraint is a propositional CNF formula over propositional variables and temporal conditions. A *solution* to a CTN is a truth value assignment to the propositional variables and temporal conditions, an assignment of values to the temporal point variables and a selection of a single relation from every qualitative temporal constraint, such that every propositional constraint and all temporal constraints whose condition is assigned *true* are satisfied simultaneously.

Using these definitions, we can intuitively divide every CTN into propositional and temporal parts. Propositional constraints impose certain restrictions on the conditions that act as a buffer and allow us to perform computations on the propositional and temporal parts of the network separately (see Figure 1).

Example 5: Consider a network with seven variables: a propositional variable P, a point variable X_1, two interval variables I_2, I_3, and three conditions C_1, C_2, C_3, with the constraints

$P \leftrightarrow C_1 \leftrightarrow C_2 \leftrightarrow C_3$,
$(C_1 : X_1\{starts\}I_2)$, $(C_2 : X_1\{finishes\}I_3)$,
$(C_3 : I_2\{before, after, meets, met{-}by\}I_3)$.

One solution is $P = C_1 = C_2 = C_3 = true$, $X_1 = 1.0$, $X_1\{starts\}I_2$, $X_1\{finishes\}I_3$, $I_2\{met{-}by\}I_3$.

The truth values of conditions control the temporal subnetwork that needs to be satisfied. Clearly as more conditions are assigned *true* the corresponding temporal network is more constrained. We can conclude:

Theorem 2: *A conditional temporal network N is consistent iff there exists a minimal model of the propositional part of N such that the set of temporal constraints whose condition is assigned true is satisfiable.*

This suggests a procedure for determining the consistency of a CTN. We enumerate all minimal models of the propositional part of a CTN and for each of them determine whether the applicable temporal network is consistent. The task of computing the minimal models of a CNF formula has been investigated and is known to be hard [4, 3]. For Horn formulas, the minimal model is unique and can be computed in polynomial time, thus tractability depends on the temporal constraints.

Corollary 1: *Given a CTN whose propositional part is Horn and temporal part is tractable, consistency can be determined in polynomial time.*

4.1 Tractable Core

We determine the consistency of $\Psi' = \Psi \cup A_1 \cup A_2 \cup A_3$ by checking the consistency of a CTN in which propositional constraints are the propositional clauses of Ψ', conditions are the temporal propositions of Ψ' and temporal constraints are specified by those temporal propositions. Axiom set A_1 introduces Horn clauses and tractable temporal constraints. Axiom set A_2 and axioms (6) and (8) of A_3 are intractable since they introduce non-Horn clauses. If we do not add axiom sets A_2 and A_3, p-satisfiability and p-entailment become tractable for Horn temporal formulas.

Theorem 3: *If Ψ is a Horn temporal formula in which every holds proposition specifies a single fluent literal and every temporal proposition specifies either single interval (if point-point) {starts} or {finishes} (if point-interval), {equals}, {before,meets},{after,met-by} or any of their disjunctions (if interval-interval), then p-consistency of $\Psi \cup A_1$ can be determined in $O(|\Psi|^2 + n_e^3)$ steps, where n_e is the number of event symbols.*

We will examine the inferences that $\Psi \cup A_1$ is capable of making. We use $\Psi \models \alpha$ to denote s-entailment and $\Psi \models_p \alpha$ to denote p-entailment. Clearly, for every sentence α, if $\Psi \cup A_1 \models_p \alpha$ then $\Psi \cup A_1 \cup A_2 \cup A_3 \models_p \alpha$ and thus $\Psi \models \alpha$. However, it might be that $\Psi \models \alpha$ and $\Psi \cup A_1 \not\models_p \alpha$. Still, if there is a clause $\alpha \rightarrow \beta$ in Ψ and $\Psi \cup A_1 \models_p \alpha$ then $\Psi \cup A_1 \models_p \beta$.

Example 6: When adding only A_1,

$$Holds(F,\ \textbf{always}) \models_p \neg Holds(\neg F,\ E_i, E_j),$$
$$Holds(F,\ \textbf{always}) \not\models_p Holds(F,\ E_i, E_j),$$
$$Holds(F_1,\ E_i, E_j) \models_p \neg Holds(\neg(F_1 \vee F_2),\ E_i, E_j),$$
$$Holds(F_1,\ E_i, E_j) \not\models_p Holds(F_1 \vee F_2,\ E_i, E_j).$$

The anomalies of the second and fourth inference can be avoided by adding some subsets of axioms A_3. In principle, as a topic for future research, it would be worthwhile to associate classes of queries with a subset of axioms A_1, A_2 and A_3 that, if added, will guarantee sound and complete inferences with respect to these queries.

Qualitative and quantitative Temporal constraint networks can be processed with a variety of algorithms, presented in [1, 6, 12, 10, 15, 13]. In particular, it was reported in [10] that qualitative temporal networks can be efficiently solved using path-consistency as a preprocessing procedure before backtracking. In [13] an effective preprocessing procedure for quantitative temporal networks is presented.

5 Conclusion

We have proposed a propositional language for temporal reasoning that is computationally effective yet is expressive enough to describe information about fluents, events and temporal constraints. The language, called `HOT`, is a set of propositional CNF formulas over *Holds, Occurs* and *Temporal* propositions as their atoms. A model (or a scenario) of an input theory determines what events happened and specifies the value of every fluent at every point in time.

We define an alternative propositional semantics for `HOT` that decouples propositional constraints from temporal constraints and allows to consider them separately. We call this framework *conditional temporal networks*. A conditional temporal network is consistent iff there exists a minimal model of the propositional constraints such that the set of temporal constraints whose condition is assigned true is satisfiable. These results render a wide variety of temporal constraint propagation techniques applicable to reasoning about events and fluents.

In particular, we identify a syntactically characterized tractable core for which a weaker (sound but incomplete) tractable inference procedure exists. This tractable core can be used as an upper bound approximation. Additional axioms yield more inferential power but at the cost of increased computational complexity. In practice, when it is known which queries are of interest, a user can add only an appropriate subset of the axioms.

Acknowledgments

We would like to thank Andre Trudel for his comments on an earlier version of this paper.

References

[1] Allen, J.F., 1983. Maintaining knowledge about temporal intervals, *CACM* 26 (11):832-843.

[2] Allen, J.F., 1984. Towards a general Theory of Action and Time, *Artificial Intelligence* 23:123-154.

[3] Ben-Eliyahu, R., Dechter, R., 1993. On computing minimal models, In Proc of AAAI-93, 2-8.

[4] Cadoli, M., 1992. On the complexity of model finding in nonmonotonic propositional logics, In *Proceedings of the Fourth Italian Conference on Theoretical Computer Science*, October 1992.

[5] Dean, T.M., McDermott, D. V., 1987. Temporal data base management, *Artificial Intelligence* 32:1-55.

[6] Dechter, R., Meiri, I., Pearl, J., 1991. Temporal Constraint Satisfaction Problems, *Artificial Intelligence* 49:61-95.

[7] Golumbic, C.M., Shamir, R., 1991. Complexity and Algorithms for Reasoning about Time: A graph theoretic approach, Rutcor Research Report 22-91 (May 1991).

[8] Kautz, H., Ladkin, P., 1991. Integrating Metric and Qualitative Temporal Reasoning, In *Proc. of AAAI-91*, pages 241-246.

[9] Kowalski, R., Sergot, M., 1986. A Logic-based Calculus of Events, *New Generation Computing* 4:67-95.

[10] Ladkin, P.B., Reinefeld, A., 1992. Effective solution of qualitative interval constraint problems, *Artificial Intelligence* 57: 105-124.

[11] McDermott, D.V., 1982. A Temporal Logic for Reasoning about Processes and Plans, *Cognitive Science* 6:101-155.

[12] Meiri, I., 1991. Combining Qualitative and Quantitative Constraints in Temporal Reasoning, In *Proc. of AAAI-91*, pages 260-267.

[13] Schwalb, E., Dechter, R., 1993. Coping with Disjunctions in Temporal Constraint Satisfaction Problems, In Proc. AAAI-93, 127-132.

[14] Shoham, Y., 1986. Reasoning about Change: time and causation from the stand point of artificial intelligence, Ph.D. dissertation, Yale Univ.

[15] Van Beek, P., 1992. Reasoning about Qualitative Temporal Information, *Artificial Intelligence* 58:297-326.

[16] Vilain, M., Kautz, H., Van Beek, P., 1989. Constraint Propagation Algorithms for Temporal Reasoning: A revised Report. In Readings in Qualitative Reasoning about Physical Systems, J. de Kleer and D. Weld (eds).