The computational complexity of satisfiability of temporal Horn formulas in propositional linear-time temporal logic

Cheng-Chia Chen and I-Peng Lin

Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, ROC

Communicated by D.A. Plaisted
Received 21 April 1992
Revised 23 November 1992

Keywords: Computational complexity; temporal logic; temporal Horn formula; temporal logic programming; satisfiability

1. Introduction

Since the invention of Prolog, a programming language based on classical first-order logic, many people have tried to extend it using similar ideas and redefine the semantics of the extended Prolog in terms of nonclassical logics [3,5,8]. The success of a programming language based on nonclassical logics usually lies in the new definition of Horn formulas and SLD-resolution-like inference rule. For modal logic and temporal logic, the corresponding definitions of Horn formulas have been available [1,4,5]. It is thus theoretically interesting to know the inherent complexity of the satisfiability problem for propositional temporal Horn formulas. In this paper we shall investigate the complexity of the satisfiability problem for two Horn fragments in propositional linear-time temporal logic. The first one contains two temporal connectives ◇ (eventuality) and □ (always) only and the second one contains an additional next-time connective ◦.

It has been shown that the satisfiability problem for linear-time temporal logic whose temporal connectives include ◇ and □ only is NP-complete and is PSPACE-complete if additional next-time connective ◦ is permitted [9]. In this paper we shall show that the complexity of the satisfiability problem remains unchanged even if the input is restricted to temporal Horn formulas. In other words, if the next-time connective is not allowed to occur in temporal Horn formulas, the problem is NP-complete; otherwise, it is PSPACE-complete.

Linear-time temporal logic without the next-time connective and the modal logic S5 are two of a few nonclassical logics whose satisfiability problems are known to be NP-complete. The NP algorithms to decide satisfiability of formulas for both logics are usually derived by the same small-model property (cf. [7,9]). It is thus natural to expect that both logics have also the same complexity when considering their Horn fragments. However, it has been proved by Farinas del Cerro and Penttonen [4] that the satisfiability of modal Horn formulas for S5 is solvable in polynomial time. Compared with the result for S5, our first complexity result for linear-time temporal logic seems surprising.

The rest of this paper is structured as follows. In Section 2 we review linear-time temporal logic and introduce temporal Horn formula. In the
next two sections we show complexity results of
the satisfiability problem for the two proposed
temporal Horn fragments.

2. Linear-time temporal logic

Syntax

The language $L$ of a propositional linear-time
temporal logic is defined on the following alphabet:

- $PV$: a finite set of propositional variables,
- classical connectives: $\neg, \vee, \wedge, \supset$,
- temporal connectives: $\Box, \Diamond, \circ$,
- parentheses: $()$.

The set of well-formed formulas of $L$ is de-
defined to be the least set of strings including $PV$
such that if $A, B$ are well-formed formulas, then
$(A \wedge B), (A \vee B), (A \supset B)$, and
$\neg A, \Diamond A, \Box A, \circ A$
are all well-formed formulas of $L$.

To avoid unnecessary parentheses, we assume
the following order of precedence for the connec-
tives:

$\circ > \Diamond > \Box > \neg > \wedge > \vee > \supset$.

Since $\wedge$ and $\vee$ are associative and commutative,
it is safe to write $A \wedge B \wedge C$ without causing any
confusion. We shall use $L^{-}$ to denote $L$ without
the temporal connective $\circ$.

Temporal Horn formula

Before we define temporal Horn formula, it
should be noticed that in the literature there is
no unified definition of temporal Horn formula
(see, for example, Temporal Prolog proposed by
Gabbay [5] and Templog proposed by Abadi and
Manna [1]). We think that the basic requirement
for a fragment to be called Horn is that each
temporal Horn formula can be translated into a
set of Horn clauses of classical first-order logic by
some satisfiability-preserving translation in poly-
nomial time (see, for example, [5] for such trans-
lation). Besides the basic requirement, in order to
be a programming language, the fragment should
also possesses an SLD-resolution-like proof
method. Our goal here, however, is not to define
a new programming language but to define a
larger class of formulas satisfying the basic re-
quirement so we can combine Templog and the
future fragment of Temporal Prolog into a unify-
ing one. Accordingly, whether there is a SLD-res-
olution-like inference rule for the definition pro-
posed is not our major concern.

The set of temporal Horn formulas $HL$ is a
subset of well-formed formulas of $L$ whose syntax
in BNF form is given as follows ("::=" and "|"
are symbols of the metalanguage). Let $p$ denote
any propositional variable and $\circ^np (n \geq 0)$ stand
for $n$ successive $\circ$'s followed by $p$.

- Temporal atom: $A ::= \circ^np (n \geq 0) | \Box B | \Diamond B$
- Body: $B ::= A | B \wedge A$
- Goal: $G ::= \neg B$
- Clause: $C ::= A | B \supset H | \Box C | \Diamond C$
- Temporal Horn formula:

$H ::= C | G | H \wedge C | H \wedge G$

For any clause of the form $B \supset H$ or of the
form $A$ where $A$ is a temporal atom, we say $B$
is the body and $H$ (or $A$) is the head of the clause.

Example 2.1. In the following formulas

$$A = \Diamond (p \wedge q) \wedge r \supset \Diamond (p \wedge q) \wedge \neg r,$$

$$B = \Box (\Diamond (p \wedge q) \wedge r \supset \Diamond (\neg p \wedge q) \wedge \neg r),$$

$$C = \Diamond (\neg p \wedge q) \supset \Diamond (p \wedge q) \wedge r,$$

$A$ and $B$ are temporal Horn formulas but $C$ is
not. The body and the head of $A$ are $\Diamond (p \wedge q) \wedge r$ and $\Diamond (p \wedge q) \wedge \neg r$, respectively; the former
is a conjunction of two atoms $\Diamond (p \wedge q)$ and $r$,
while the latter is a conjunction of one temporal
atom $\Diamond (p \wedge q)$ and one goal $\neg r$. $C$ is not a
temporal Horn formula because $\Diamond (\neg p \wedge q)$ is
not a body.

The set of temporal Horn formulas defined
above is essentially the future fragment of Tem-
poral Prolog augmented with the next-time con-
nective $\circ$. The major differences are (1) we do
not have the past-time connective $P$ and have an
additional next-time connective $\circ$ and (2) Tempo-
ral Prolog does not allow the “always” connective
$\Box$ to appear in bodies of clauses. Although the
definitions are distinct, our first complexity result for the Horn fragment not allowing occurrences of $\circ$ also holds for his definition. As for the definition of Templog proposed by Abadi and Manna, which permits the use of the next-time connective but does not allow the $\circ$ connective to appear in heads, our second theorem is still valid.

In the following, we shall use $HL^-$ to denote $HL$ without the $\circ$ connective.

**Semantics**

A linear-time structure $S$ for a propositional temporal language $L$ is a pair $(s, \xi)$, where $s = (s_0, s_1, \ldots)$ is an $\omega$-sequence of states and $\xi$ is a mapping from $(s_0, \ldots)$ to the power set of $PV$. An interpretation is a pair $(S, t)$ where $S$ is a structure and $t$ is a state in $s$. The truth of a formula in an interpretation is defined inductively as follows. Let $p$ be any propositional variable, $A$ and $B$ be any formulas.

- $S, s_i \models p$ iff $p \in \xi(s_i)$.
- $S, s_i \models A \land B$ iff $S, s_i \models A$ and $S, s_i \models B$.
- $S, s_i \models \neg A$ iff $S, s_i \not\models A$.
- $S, s_i \models \diamond A$ iff $\exists j > i$ such that $S, s_j \models A$.
- $S, s_i \models \square A$ iff $S, s_{i+1} \models A$.

The other connectives $\lor$, $\Rightarrow$ and $\land$ are regarded as defined operators and are defined as usual. In particular, $\square A$ is defined as $\neg \diamond \neg A$, so we have

- $S, s_i \models \square A$ iff $S, s_j \models A$ for all $j \geq i$.

We say a formula $A$ is satisfiable if there exists a structure $S$ such that $S, s_0 \models A$.

The following propositions about linear-time temporal logic are fundamental to our results.

**Proposition 2.2.** The problem that, given a formula in $L^-$, whether there is a linear-time structure satisfying it, is NP-complete [9].

**Proposition 2.3.** The problem that, given a formula in $L$, whether there is a linear-time structure satisfying it, is PSPACE-complete [9].

3. NP-completeness result for $HL^-$

We first consider the Horn fragment in which $\circ$ does not occur.

**Theorem 3.1.** The satisfiability problem for linear-time temporal logic with the input restricted to $HL^-$ is NP-complete.

**Proof.** Since $HL^-$ is a subset of $L^-$, with Proposition 2.2, it suffices to prove that the satisfiability problem for $HL^-$ is NP-hard. We will reduce the satisfiability problem for classical propositional clauses to the satisfiability of temporal Horn formulas in $HL^-$ in polynomial time.

Assume the input is a set of propositional clauses $\mathcal{C} = \{C_1, \ldots, C_m\}$ where $m > 1$ and each $C_i$ (where $1 \leq i \leq m$) is a set of literals that does not contain complementary literals.

Let $V_{\mathcal{C}} = \{X_1, \ldots, X_n\}$ be the set of propositional variables appearing in $\mathcal{C}$. The set of propositional variables $PV$ used in $MT(\mathcal{C})$ includes $V_{\mathcal{C}}$ and the set $V_{\mathcal{C}} = \{X | X \in V_{\mathcal{C}}\}$, each element $X$ of which is a new variable not used before and is intended to stand for the complement of $X$.

Let $MT(\mathcal{C}) = \bigwedge_{1 \leq i \leq 3} T_i$, where each $T_i$ is given as follows.

- $T_1 = \bigwedge_{1 \leq i \leq n}(\diamond X_i \land \diamond \overline{X}_i)$.
- For $1 \leq i \leq n$, let $g_i = \diamond (X_i \land \diamond \overline{X}_i)$ and $\overline{g}_i = \diamond (\overline{X}_i \land \diamond X_i)$. Let $T_2 = \bigwedge_{1 \leq i \leq n}(g_i \land \overline{g}_i)$.
- For each $1 \leq i \leq m$, $1 \leq j \leq p_i$, let

$$l_{ij} = \begin{cases} \overline{g}_k & \text{if } L_{ij} = X_k \\ g_k & \text{if } L_{ij} = \neg X_k \end{cases}$$

Then let $T_3 = \bigwedge_{1 \leq i \leq m} (l_{i1} \land \cdots \land l_{ip_i})$.

**Example 3.2.** Let $\mathcal{C} = \{C_1, C_2, C_3\}$, where $C_1 = \{X_1, \neg X_2, X_4\}$, $C_2 = \{\neg X_1, X_2, X_3\}$, $C_3 = \{\neg X_1, \neg X_3\}$.

Then $m$ (the number of clauses in $\mathcal{C}$) = 3, $n$ (the number of propositional variables in $\mathcal{C}$) = 4, and

$$T_1 = \diamond X_1 \land \diamond \overline{X}_1 \land \diamond X_2 \land \diamond \overline{X}_2$$

$$\land \diamond X_3 \land \diamond \overline{X}_3 \land \diamond X_4 \land \diamond \overline{X}_4.$$
The formula $MT(g)$ defined above is clearly a temporal Horn formula; it is also easy to see that $MT(B)$ can be constructed from $B$ in time polynomial in the size of $S$.

To understand $MT(\mathcal{E})$, let $S$ be any linear-time structure satisfying $T_1$ and $T_2$ and suppose that we use each $X_i$ (and $\overline{X_i}$) to denote an event which may and may not occur at any state of the time sequence. We say event $X$ occurs before event $Y$ if there are two states $t_i, t_j$ in $S$ such that $X$ occurs at $t_i$, $Y$ occurs at $t_j$ and $i < j$. Then, $T_1$ means every event must occur at least once at the time sequence; $g_i$ says $X_i$ occurs before $\overline{X_i}$ and $\overline{g}_i$ says $\overline{X_i}$ occurs before $X_i$, so $T_2$ means that it is not the case that $X_i$ occurs before $\overline{X_i}$ and $\overline{X_i}$ occurs before $X_i$ for all $i < n$. $T_1$, $T_2$ and the linear-order property of the time sequence imply that for each $i \leq n$, exactly one of $g_i$ and $\overline{g}_i$ is true at the initial state of any linear-time structure satisfying $MT(\mathcal{E})$.

Lemma 3.3. Let $S = (s, \xi)$ be any linear-time structure with $s = s_0, s_1, \ldots$ such that $S, s_0 \models MT(\mathcal{E})$. Then, for any $1 \leq i \leq n$, $S, s_0 \models g_i$ iff $S, s_0 \not\models g_i$.

Proof. By $T_1$, $S, s_0 \models \diamond X_i \land \diamond \overline{X_i}$. Therefore, there must exist $u \leq v \in s$ such that $X_i \in \xi(s_u)$ and $\overline{X_i} \in \xi(s_v)$.

If $u \leq v$, $S, s_u \models X_i \land \diamond \overline{X_i}$, so $S, s_0 \models g_i$; likewise, if $v < u$, $S, s_v \models \overline{X_i} \land \diamond X_i$, so $S, s_0 \models \overline{g}_i$. Consequently, either $g_i$ or $\overline{g}_i$ must be true at $s_0$.

However, $T_2$ says at least one of $g_i$ and $\overline{g}_i$ must be false at $s_0$. Therefore, $S, s_0 \models g_i$ iff $S, s_0 \not\models \overline{g}_i$. □

Proof of Theorem 3.1 (continued). Lemma 3.3 provides us a natural way to establish connections between truth assignments on $\mathcal{E}$ and linear-time structures satisfying $T_1$ and $T_2$. For each linear-time structure $S$ satisfying $T_1$ and $T_2$, we associate with it a truth assignment $I(S)$ on $\mathcal{E}$ in which $X_i$ is assigned true (false) if $g_i$ ($\overline{g}_i$) holds at the initial state of $S$. It is clear from Lemma 3.3 that $I(S)$ assigns a unique truth-value to each $X$ in $\mathcal{E}$. It is also easy to see that for each truth assignment $I$ on $\mathcal{E}$, there is a structure $S$ satisfying $T_1$ and $T_2$ such that if $I$ assigns true (false) to $X_i$, then $g_i$ ($\overline{g}_i$) holds at the initial state of $S$.

Now we might use $g_i$ and $\overline{g}_i$ to encode $X_i$ and $\overline{X_i}$, respectively; we would then obtain a temporal formula which is satisfiable if and only if $\mathcal{E}$ is satisfiable. But indeed we cannot use the formula since it is not a temporal Horn formula. Fortunately, by Lemma 3.3, $g_i$ and $\overline{g}_i$ have the same truth value at the initial state of any structure satisfying $T_1$ and $T_2$, so instead of using $g_i$ and $\overline{g}_i$, we use $\overline{g}_i$ and $\overline{g}_i$ to encode $X_i$ and $\overline{X_i}$, respectively, and $T_3$ is the resulting temporal Horn formula with the negation $\neg$ factored out.

The following lemma, which formalizes the above discussion and states that the transformation from $\mathcal{E}$ to $MT(\mathcal{E})$ is satisfiability-preserving, together with the fact that $MT(\mathcal{E})$ can be obtained from $\mathcal{E}$ in polynomial time, implies that the satisfiability problem of temporal Horn formulas in $HL^-$ is NP-hard. We thus obtain Theorem 3.1. □

Lemma 3.4. $\mathcal{E}$ is satisfiable iff $MT(\mathcal{E})$ is satisfiable in linear-time temporal logic.

Proof. ($\Rightarrow$) Since $\mathcal{E}$ is satisfiable, there exists a literal set $E = \{L_1, \ldots, L_n\}$ where $L_i$ is either $X_i$ or $\overline{X_i}$ such that, for each $\mathcal{E}' \in \mathcal{E}$, $E \cap \mathcal{E}' \neq \emptyset$.

Now let the structure $S = (s, \xi)$ with $s = s_0, s_1, \ldots$ and $\xi$ defined as follows:

1. $\xi(s_0) = \{X_i \mid X_i \in E\} \cup \{\overline{X_i} \mid \overline{X_i} \in E\}$
2. $\xi(s_i) = \{X_i \mid X_i \in E\} \cup \{\overline{X_i} \mid \overline{X_i} \in E\}$
3. $\xi(s_i) = \xi(s_j)$ if $j > 1$.

It can be easily verified that $S, s_0 \models MT(\mathcal{E})$.

($\Leftarrow$) Let $S = (s, \xi)$ be any structure such that
From $S,$ we construct a literal set $E = \{L, \ldots, L\},$ where

$L_i = \begin{cases} X_i & \text{if } S, s_0 = g_i, \\ \neg X_i & \text{if } S, s_0 = \bar{g}_i. \end{cases}$

By Lemma 3.3, for $1 \leq i \leq m,$ exactly one of $X_i$ and $\neg X_i$ belongs to $E.$

Now we show that $E \cap C \neq \emptyset$ for each $C \in \mathcal{G},$ so $\mathcal{G}$ is satisfiable. Let

$C = \{X_{i_1}, \ldots, X_{i_\alpha}, \neg X_{i_{\beta}}, \ldots, \neg X_{i_{\gamma}}\}$

be any clause in $\mathcal{G}$ where $\alpha \geq 0, \beta \geq 0.$

By $T_3,$ we have

$S, s_0 = \neg \bigwedge_{1 \leq i \leq \alpha} \bar{g}_{u_i} \land \bigwedge_{1 \leq i \leq \beta} g_{v_i}.$

So either $S, s_0 = \neg \bar{g}_{u_i}$ for some $1 \leq i \leq \alpha,$ or $S, s_0 = \neg g_{v_i}$ for some $1 \leq i \leq \beta.$ By Lemma 3.3, either $S, s_0 = \bar{g}_{u_i}$ or $S, s_0 = g_{v_i}.$ Accordingly, either $X_{i_j} \in E$ or $\neg X_{i_j} \in E,$ and $E \cap C \neq \emptyset.$ \hfill \Box

4. PSPACE-completeness result for HL

**Theorem 4.1.** The satisfiability problem for linear-time temporal logic with the input restricted to HL is PSPACE-complete.

**Proof.** Since PSPACE is closed under complementation [6], with Proposition 2.3 available, it suffices to show that the unsatisfiability problem of temporal Horn formulas is PSPACE-hard. We will reduce the recognition of a language $L(M)$ accepted by a deterministic single-tape one-way Turing machine $M$ which operates in polynomial space $s(n) > n$ to the unsatisfiability problem of temporal Horn formulas. Without loss of generality, assume $s(n)$ is computable in time polynomial of $n,$ $M$ halts of all inputs (see Lemma 12.1 of [6] for details) and never run outside the first $s(n)$ tape cells where $n$ is the size of the input.

Let $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \# \rangle,$ where $Q$ is the finite set of states, $\Gamma$ is the set of allowed tape symbols, $\# \in \Gamma$ is the blank symbol, $\Sigma \subset \Gamma$ is the set of input symbols, $\delta$ is the next move function, a partial mapping from $Q \times \Gamma$ to $Q \times \Gamma \times \{L, R\},$ $q_0 \in Q$ is the initial state, and $q_f \in Q$ is the unique final state.

Now, for each string $x = a_1 \cdots a_n$ of length $n > 0,$ a temporal Horn formula $MT_M(x)$ is generated as follows.

1. The set of propositional variables $PV$ appearing in $MT_M(x)$ includes

2. $C = \{C_{ia} | 0 \leq i \leq s(n), a \in \Gamma\}$ whose intended usage is that $C_{ia}$ is true iff the $i$th tape cell contains symbol $a.$

3. $ID_0 = \{q_0, P_1, C_{0\#}, C_{1a_1}, C_{2a_2}, \ldots, C_{na_n}, C_{(n+1)a}, \ldots, C_{s(n)a}\}.$ $ID_0$ is the set of temporal atoms used to describe the initial configuration of the Turing machine with the input $x$ given.

4. $ID_f = \{-a_j\}.$ $ID_f$ states the impossibility of reaching the final state.

5. The transitions between successive configurations are encoded by the set of clauses $Tr = Tr_1 \cup Tr_2 \cup Tr_3,$ where

(a) $Tr_1 = \{\square (P_i \land C_{ia} \supset \circ C_{ja}) \mid 0 \leq i \leq s(n), j \neq i, a \in \Gamma\},$

(b) $Tr_2 = \{\square ((P_i \land C_{ia} \land q_1) \supset \circ (P_{i+1} \land C_{ib} \land q_2)) \mid \delta(q_1, a) = (q_2, b, R), 0 \leq i < s(n)\},$

(c) $Tr_3 = \{\square ((P_i \land C_{ia} \land q_1) \supset \circ (P_{i-1} \land C_{ib} \land q_2)) \mid \delta(q_1, a) = (q_2, b, L), 0 < i \leq s(n)\}.$

$Tr_1$ says that the contents of the tape cells remain unchanged in the next configuration if they are not currently pointed to by the tape head; $Tr_2$ and $Tr_3$ describe to what symbol the current scanned tape cell should be changed, to what position the tape head should point and to what state the control store should move in the next configuration. Note, more formally, in order
to be temporal Horn clauses, the clauses in $Tr_2$ and $Tr_3$ should have their $\circ$ operators appearing in the heads moving inside the conjunctions. However, we deliberately write them in "non-standard" form so we can capture their intuitive meaning.

(5) Finally, let $MT_M(x)$ be the conjunction of all formulas in $ID_0 \cup ID_t \cup Tr$.

With the given Turing machine $M$ and the polynomial $s(n)$ fixed, it is obvious that $ID_0$ and $ID_t$ require $s(n) + 3$ clauses and 1 clause, respectively, and $Tr_1$ and $Tr_2 \cup Tr_3$ require $(s(n+1) \times s(n) \times |\Gamma|)$ clauses and $s(n)$ clauses, respectively. Since the size of each clauses in $MT_M(x)$ is less than a constant $c$, the total size of $MT_M(x) = O(s^2(n))$. Hence it is easy to see that $MT_M(x)$ can be generated in polynomial time.

We claim that $x$ is accepted by $M$ iff $MT_M(x)$ is unsatisfiable.

For the proof of the only-if part, we note that $x \in L(M)$ implies that the computation of $M$ on $x$ terminates at the final state $q_f$ using no more space than $s(|x|)$. Hence $\Diamond q_f$ is a consequence of $ID_0 \cup Tr$. However, $ID_t$ contains $\neg \Diamond q_f$, so $MT_M(x)$ is unsatisfiable.

On the other hand, if $x \notin L(M)$, since Turing machine $M$ halts on every input, $M$ must terminate at a state $q_i \neq q_f$ at some step $m$ and for any $j < m$, the state of $M$ at step $j$ is not the final state $q_f$. Let $\beta_0, \beta_1, \ldots, \beta_m$ be the computation of $M$ on $x$. We construct a linear-time structure $T = (t, \xi)$ with $t = \{t_0, t_1, \ldots\}$ and $\xi$ defined as follows.

- For $0 \leq i \leq m$, $P_i \in \xi(t_i)$ iff the tape head points to tape cell $j$ at configuration $\beta_i$, $a \in \xi(t_i)$ iff the control store is at state $q \at \beta_i$, $C_{ja} \in \xi(t_i)$ iff the $j$th tape cell contains symbol $a$ at $\beta_i$.
- For $i > m$, $\xi(i) = \xi(m)$.

It can be easily verified that for any $i \in \omega$ and for any always-clause $\square(l \circ r) \in Tr$, either $T, t_{i-1} \models l$ or $T, t_i \models r$. Thus $Tr$ is valid in $T$. Moreover, $\neg \Diamond q_f$ is satisfied by any $t_i$ because the control store is not at $q_i$ for any $\beta_i$. Finally, it is also easy to see that $\xi(t_0) = ID_0$, we thus have $T, t_0 \models MT_M(x)$. $\square$

It should be noticed that each formula in $Tr_2$ or in $Tr_3$ indeed is not a temporal clause of Templog. However, we can split every clause of the form $\square((l \circ \circ p_i \land p_i \land p_j))$ in $Tr_2 \cup Tr_3$ into three templog clauses $\square((l \circ \circ p_i)) (i = 1, 2, 3)$. The resulting set of formulas then is a set of templog clauses, so Theorem 4.1 is valid for the temporal Horn formula used in Templog as well.

References