THE COMPLEXITY OF PROPOSITIONAL LINEAR TEMPORAL LOGICS

A. P. Sistla and E. M. Clarke
Aiken Computation Laboratory
Harvard University
Cambridge, MA 02138

Abstract

We consider the complexity of satisfiability and determination of truth in a particular finite structure for different propositional linear temporal logics. We show that both the above problems are NP-complete for the logic with F operator and are PSPACE-complete for the logics with F,X, with U, with U,S,X, and Wolper's extended logic with regular operators [Wo81].

1. Introduction

Linear Temporal Logic was introduced by Pnueli [Pn77] as an appropriate formal system for reasoning about parallel programs. This logic permits the description of a program's execution history without the explicit introduction of program states or time. Moreover, important correctness properties such as mutual exclusion, deadlock freedom, and absence of starvation can be elegantly expressed in this system. Proving that a parallel program satisfies some correctness property consists of deducing the formula for that property from *program axioms* which characterize the possible interleaving of atomic statements of the individual processes. An important special case occurs when the program is finite state. In this case, the program axioms and correctness specification can be expressed in the *propositional* version of the logic and provability becomes *decidable*. A number of researchers (e.g., [MW81]) have attempted to use such a decision procedure for constructing correct finite-state programs.

In this paper we examine the inherent complexity of decision procedures for *validity*, *satisfiability*, and *truth in a particular structure* for propositional logics with the temporal operators F (eventually), G (globally), X (next-time), U (until) and S (since). We first consider the logic L(F) in which F is the only temporal operator. We prove a *linear size model theorem* from which a nondeterministic polynomial time bounded decision procedure for satisfiability can be obtained. It immediately follows that satisfiability is NP-complete for L(F). This result is surprising since it shows that the set of satisfiable formulae in L(F) is no higher in the complexity hierarchy than the set of satisfiable formulas of ordinary propositional logic. The proof of the linear size model theorem also provides insight into the expressive power of the F operator. A similar NP-completeness result has been obtained by Ladner [La77] for the modal logic S5. However, he uses an entirely different technique to show the existence of small models.

Next, we give a polynomial space bounded decision procedure for satisfiability of formulae L(U,S,X). The decision procedure given for this problem in [Wo81] is tableaux based and requires

This research was supported by NSF grant MCS79-08365.

exponential space. Nor does our result follow from the well-known translation [GPSS81] of temporal formulas into formulas of the first order language with the linear ordering $<$ and monadic predicates $p_1,\ldots,p_n,\ldots$. If $<$ is interpreted as the ω-ordering, then determination of the satisfiable formulas in this language is at least exponential-time hard.

We also show that satisfiability for L(F,X) is PSPACE-complete for the logics L(F,X), L(U), L(U,X), L(U,S,X) and Wolper's extended logic [Wo81]. These results are also surprising because all of these logics have different expressive powers (some are more powerful than others) and because the satisfiability problem for branching time logics with these operators is EXPTIME-complete.

Finally, we consider the question of whether a temporal formula is true in a *particular* structure. Since we are interested in modelling the execution behavior of programs, we restrict our attention to structures (called R-*structures*) in which the state histories are generated by a binary relation on the state set. For this class of structures, the determination of truth is NP-complete for L(F) but PSPACE-complete for L(F,X), L(U) and L(U,S,X). The corresponding problem for branching-time logics has been shown to be in P [CE81].

The paper is organized as follows: Section 2 defines the syntax and semantics of the linear temporal logic that we use in the remainder of the paper. In Section 3 we prove the linear size model theorem for L(F) and the corresponding NP-completeness results. Section 4 contains the PSPACE-completeness results for L(F,X), L(U), and L(U,S,X). In Section 5 we show how our results can be extended to Wolper's Logic.

2. Notation and Basic Definitions

We use the following convention for symbols:

P,Q,R,...	denote atomic formulae.
f,g,h,...	denote formulae.
s,t,u,...	denote finite or infinite sequences. We always assume $s = (s_0,s_1,\ldots)$.
S,T,W,...	denote structures.

If $O_1,\ldots,O_k \in \{X,F,G,U,S\}$ are distinct operators then $L(O_1,\ldots,O_k)$ denotes the propositional temporal logic restricted to these operators, e.g. L(F,G), L(X,F,G), etc. A formula f in L(X,U,S,F,G) is inductively defined as follows:

If P is an atomic proposition then P is a formula;

If f_1,f_2 are formulas then $f_1 \wedge f_2$, $\neg f_1$, Xf_1, f_1Uf_2, f_1Sf_2, Ff_1, Gf_1 are also formulas.

Length(f) is the number of symbols in formula f; SF(f) denotes the set of subformulas of f.

Let $\mathcal{P}$ be a finite set of atomic propositions. We assume that all the atomic propositions in the formulae we consider are from $\mathcal{P}$. A *structure* $S = (s,\xi)$ is a pair, where s is an infinite sequence of distinct states denoted by $s = (s_0,s_1,\ldots)$ and $\xi:\{s_0,s_1,\ldots\} \to 2^{\mathcal{P}}$. Intuitively ξ specifies which atomic propositions are true in each state. For notational convenience we assume that the states in the sequences are from a fixed set and are all distinct. An *interpretation* is a pair $<S,s_j>$ where $S = (s,\xi)$ is a structure and s_j appears in s. We define the truth of a formula f in an interpretation $<S,s_j>$ (denoted by $S,s_j \models f$) inductively as follows:

$S,s_j \models P$	iff	$P \in \xi(s_j)$;
$S,s_j \models f_1 \wedge f_2$	iff	$S,s_j \models f_1$ and $S,s_j \models f_2$;
$S,s_j \models \neg f_1$	iff	$S,s_j \not\models f_1$;
$S,s_j \models Ff_1$	iff	$\exists k \geq j$ such that $S,s_k \models f_1$;
$S,s_j \models Gf_1$	iff	$\forall k \geq j$ $S,s_k \models f_1$;
$S,s_j \models Xf_1$	iff	$S,s_{j+1} \models f_1$;
$S,s_j \models f_1Uf_2$	iff	$\exists k \geq j$ such that $S,s_k \models f_2$ and $\forall i$ such that $j \leq i < k$, $S,s_i \models f_1$;
$S,s_j \models f_1Sf_2$	iff	$\exists k \leq j$, $S,s_k \models f_2$ and $\forall i$ such that $k \leq i < j$, $S,s_i \models f_1$.

Note that $Ff \equiv \text{true } U f$ and $Gf \equiv \neg F(\neg f)$. A formula $f \in L$, is *satisfiable* iff there exists a structure $S = (s,\xi)$ such that $S,s_i \models f$ for some i; f is *valid* iff for all structures $S = (s,\xi)$ and for all $i \geq 0$, $S,s_i \models f$.

An R-structure T is a triple (N,R,η), where N is a finite set of states; $R \subseteq N \times N$

is a total binary relation (that is $\forall t \in N\ \exists t' \in N$ such that $(t,t') \in R$), and $\xi: N \to 2^{\mathcal{P}}$. A *path* p in T is an infinite sequence $(p_0,p_1,\ldots)$ where $\forall i \geq 0$, $p_i \in N$, and $(p_i,p_{i+1}) \in R$. Throughout this paper, for a path p in a R-structure $T = (N,R,\eta)$ we let S_p denote the structure (s,ξ) where $\forall i \geq 0$, $\xi(s_i) = \eta(p_i)$.

The global behavior of a finite state parallel program can be modelled as an R-structure. In the R-structure each path starting from the initial state represents a possible interleaving of executions of the individual processes in the program. In many cases, the correctness requirements of the concurrent system can be expresses by a formula of propositional linear time logic. The system will be correct iff every possible execution sequence satisfies this formula; i.e., every path beginning at the initial state in the corresponding R-structure satisfies the formula. For these reasons the following problem (which we call *the determination of truth in a R-structure*) is important in verifying finite state parallel programs:

Given a R-structure T, a state $p_0 \in N$, a formula $f \in L$, is there a path p in T starting from p_0 such that $S_p,s_0 \models f$?

3. The Complexity of L(F)

Let $S = (s,\xi)$ be a structure and let $s'' = (s_j,s_{j+1},\ldots)$ be the maximal suffix of s such that for each s_k in s" the following condition holds:

$\forall \ell\ \exists i$ such that $i > \ell$ and $\xi(s_i) = \xi(s_k)$,

that is, there exist infinitely many states in s" which have the same assignment of atomic propositions, as s_k. It is easily seen that such an s" exists (because $\mathcal{P}$ is finite), and s" is unique. Let $s = s' \circ s''$. Define *init(s)* = s', *final(s)* = s", *range(s)* = $\{\xi(s_k) \mid s_k$ is in s"$\}$ and *size(s)* = length(init(s)) + card(range(s)). Thus, range(s) is the set of all assignments of atomic propositions which occur infinitely often in s. Note that init(s) is a finite sequence (and can be the null sequence!), final(s) is an infinite sequence, and range(s) is a subset of $2^{\mathcal{P}}$.

THEOREM 3.1. (Linear size model theorem for L(F)). *If* $f \in L(F)$ *is satisfiable then there exists a structure* $S = (s,\xi)$ *such that* size(s) $\leq$ 2 * length(f) *and* $S,s_0 \models f$.

Proof of Theorem 3.1 is based on the following lemmas which provide insight on the expressive power of the F operator.

LEMMA 3.2 *Let* $S = (s,\xi)$ *be a structure and let* s_j, s_k *be states in* final(s) *such that* $\xi(s_j) = \xi(s_k)$; *then for all* $f \in L(F)$, $S,s_j \models f$ *iff* $S,s_k \models f$. □

This lemma shows that all states in final(s) with the same assignment of atomic propositions satisfy the same formulae in L(F).

LEMMA 3.3. *Let* $S = (s,\xi)$, $T = (t,\pi)$ *be structures such that* length(init(s)) = length(init(t)); *for all* $j <$ length(init(s)) $\xi(s_j) = \pi(t_j)$; $\xi(s_0) = \pi(t_0)$ *(this is necessary for the case when length(init(s)) = 0) and* range(s) = range(t); *then for all* $f \in L(F)$, $S,s_0 \models f$ *iff* $T,t_0 \models f$. □

It follows from Lemma 3.3 that formulas in L(F) cannot distinguish the order of occurrence of states in final(s).

Let $s = (s_0,\ldots)$, $t = (t_0,t_1,\ldots)$ be finite or infinite sequences with all states in s,t being distinct. t *is a subsequence of* s (*written* $t \leq s$) iff there exist integers $i_0,i_1,\ldots$ s.t. $i_0 < i_1 < i_2 < \ldots$ and for all $j \geq 0$, $s_{i_j} = t_j$. Let $S = (s,\xi)$ be a structure. t is an *acceptable subsequence of* s (*written* $t \sqsubset s$ or $s \sqsupset t$) if $t \leq s$ and if any s_j in final(s) is contained in t, then all states s_k in final(s) such that $\xi(s_k) = \xi(s_j)$ are also contained in t. Init(t), final(t), range(t), size(t) are also appropriately defined. If $t \sqsubset s$ then size(t) $\leq$ size(s). Note that if $t \leq s$, then it is possible that size(t) > size(s).

LEMMA 3.4. *Let* $S = (s,\xi)$ *be a structure and let* $t \sqsubseteq s$ *be such that for all* $j \geq 0$, $S, t_j \models f$ *where* $f \in L(F)$. *Then*

(a) *there exists an infinite sequence* u *such that* $u \sqsubseteq s$, $size(u) < c * length(f)$ *and*

(b) *for all structures* $W = (\omega,\xi)$ *where* $u \sqsubseteq \omega \sqsubseteq s$, ζ *is the restriction of* ξ *to the states in* ω, *the following condition holds:*

For any i *if* ω_i *is present in* t *then* $W,\omega_i \models f$.

Proof Sketch of Lemma 3.4.

Using Demorgan's laws and the identities $\neg Ff = G\neg f$, $\neg Gf = F\neg f$, any $f' \in L(F)$ can be converted to an equivalent formula f in which all negations apply to atomic propositions only. For formulas of this kind we prove lemma 3.4 with $c = 1$. The proof is by induction on the length of the formula.

Basis: $f = P$ or $\neg P$. u = null sequence satisfies the lemma.

Induction: (i) $f = f_1 \wedge f_2$.
For all $t_j \in t$, $S,t_j \models f_1$ and $s,t_j \models f_2$. By induction hypothesis there exist u_1,u_2 such that $size(u_1) < length(f_1)$, $size(u_2) < length(f_2)$, and (b) holds for u_1,f_1 and u_2,f_2. Let $u \sqsubseteq s$ be the sequence containing the states of u_1 and u_2. $Size(u) \leq size(u_1) + size(u_2) < length(f)$, and (b) holds for u,f.

(ii) $f = f_1 \wedge f_2$ - argument is similar as in (i)

(iii) $f = Ff_1$:

Case 1: t is finite. Let t_n be the last state of t. $S,t_n \models Ff_1$. Hence there is a s_j appearing after t_n in s such that $S,s_j \models f_1$. If s_j is in init(s), then let $t' = (s_j)$; otherwise let t' be the subsequence of all states s_k in final(s), such that $\xi(s_k) = \xi(s_j)$. For all s_k in t', $S,s_k \models f_1$. By induction hypothesis there is a $u' \sqsubseteq s$ such that $size(u') < length(f_1)$ and (b) is satisfied for u', f_1 and with $t = t'$. Now let $u \sqsubseteq s$, be the sequence containing all states of u' and t'. Then $size(u) \leq size(t') + size(u') = 1 + size(u') < length(f)$ and (b) holds.

Case 2: t is infinite. There exist infinitely many k such that $S,s_k \models f_1$. Let $t' = (t'_0,\ldots) \sqsubseteq s$ be such that t' is infinite, for all $j \geq 0$ $\xi(t'_j) = \xi(t'_{j+1})$, and $S,t'_j \models f_1$. The remainder of the argument is as in case 1.

(iv) $f = Gf_1$: If t_0 is in init(t) then let t' = suffix of s starting from t_0, otherwise let $t' = final(s)$. Clearly for all $j \geq 0$, $S, t'_j \models f_1$. By induction hypothesis there is a $u' \sqsubseteq s$ such that $size(u') < length(f_1)$ and (b) holds for u',f_1 and with $t = t'$. Since t' is a suffix of s, it is easily observed that (b) holds for u,f, and t.

Since any formula $f \in L(F)$ can be converted into an equivalent formula in which negations are applied to atomic propositions only, and whose length is no more than double the length of the original formula, we see that Lemma 3.4 is true with $c = 2$.

Proof Sketch of Theorem 3.1. Assume f is satisfiable and let $V = (v,\phi)$ be a structure such that $V,v_0 \models f$. Let t be the sequence as follows. If $length(init(v)) > 0$ then $t = (v_0)$; otherwise t is the sequence containing all states v_i such that $\phi(v_i) = \phi(v_0)$. Clearly $t \sqsubseteq v$, and due to Lemma 3.2 for all $i \geq 0$, $V, t_i \models f$. Now applying Lemma 3.4 with $S = V$, t we get an infinite sequence $u \sqsubseteq v$, such that $size(u) < 2 * length(f)$ and u satisfies the condition given in Lemma 3.4. Let $s \leq v$ be the sequence containing all the states of t and u. Then $s \sqsubseteq v$ and $size(s) \leq size(t) + size(u) = 1 + size(u) \leq 2 * length(f)$. Let $S = (s,\xi)$ where ξ is the restriction of ϕ to the states appearing in s. Then from Lemma 3.4, $S,s_0 \models f$.

THEOREM 3.5. *The following problems are NP-complete for the linear time logic* L(F).

(i) *Determination of truth in a* R-*structure.*

(ii) *Satisfiability.*

Proof. (i) We will prove that determining truth in an R-structure is NP-hard by reducing 3-SAT to this problem. Let $g = C_1 \wedge C_2 \wedge \ldots \wedge C_m$

be a boolean formula in 3-CNF where $C_i = \ell_{i1} \vee \ell_{i2} \vee \ell_{i3}$ (for $1 \le i \le m$), $\ell_{ik} = x_j$ or $\neg x_j$ $(1 \le k \le 3)$ for some j such that $1 \le j \le n$. $x_1, x_2, \ldots, x_n$ are the variables appearing in g. Let $T = (N,R,\eta)$ be the R-structure defined as follows:

$$\mathscr{P} = \{\tilde{C}_i \mid 1 \le i \le m\}$$

T can be described by the graph shown below:

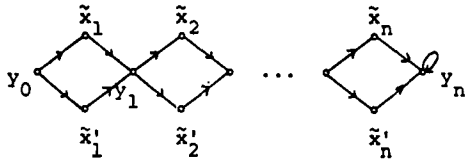

Figure 1

$$N = \{\tilde{x}_i \mid 1 \le i \le n\} \cup \{\tilde{x}'_i \mid 1 \le i \le n\} \cup \{y_i \mid 0 \le i \le n\}$$

$$R = \{(y_{i-1}, \tilde{x}_i), (y_{i-1}, \tilde{x}'_i), (\tilde{x}_i, y_i), (\tilde{x}'_i, y_i) \mid 1 \le i \le n\} \cup \{(y_n, y_n)\}$$

$\eta(\tilde{x}_i) = \{\tilde{C}_j \mid x_i$ appears as a literal in C_j i.e., for some k $1 \le k \le 3$, $\ell_{jk} = x_i\}$

$\eta(x'_i) = \{C_j \mid \neg x_i$ appears as a literal in $C_j\}$

$\eta(y_j) = \emptyset \qquad (0 \le j \le m)$.

It can easily be proved that g is satisfiable iff there exists a path p in T starting from y_0 such that $(S_p, s_0) \models F\tilde{C}_1 \wedge F\tilde{C}_2 \wedge \ldots \wedge F\tilde{C}_m$. The above reduction is a polynomial reduction. Hence determination of truth in a R-structure is NP-hard for the language L(F).

Let $T = (N,R,\eta)$ be an R-structure. Any path p in T can be uniquely decomposed into p', p'' such that $p = p' \circ p''$, any state that appears in p'' appears in it infinitely often, and p'' is the maximal such suffix. All the states in p'' belong to a strongly connected component in the graph of T. Using Lemma 3.4 it can be shown if there is a path q in T starting from q_0 such that $(S_q, s_0) \models f$, then there exists a path p in T starting from q_0 such that $(S_p, s_0) \models f$, $p = p' \circ p''$, and length(p') $\le$ length(f)$\cdot$ card(N). A nondeterministic TM M guesses p' and the set C of states appearing in p''. Next, it verifies that p' is a path starting from q_0 in T, that the subgraph containing nodes of C is strongly connected, and that there is an edge from the last state of p' to a state in C. Then M uses the following algorithm to verify if $(S_p, s_0) \models f$. M labels each node x in p' or C with subformulae of f as follows:

```
For each node x  label(x) ← ∅;
For each formula g ∈ SF(f) in the increasing
order of length(g) do
   For each node x in p' or C do
      CASE g
         g = P:   If P ∈ η(x) then
                  label(x) ← label(x) ∪ {P};
         g = ¬g:  If g1 ∉ label(x) then
                  label(x) ← label(x) ∪ {g};
         g = Fg1: If g1 ∈ label(y) for some
                  y ∈ C then
                  label(x) ← label(x) ∪ {g};
                  If x is in p' and there
                  is a state y in p' after
                  x such that g1 ∈ label(y)
                  then label(x) ← label(x) ∪ {g};
         g = g1 ∧ g2: If g1, g2 ∈ label(x) then
                  label(x) ← label(x) ∪ {g};
      End Case:
   End for;
End for;
Accept iff f ∈ label(q0).
```

It can easily be shown that the above algorithm works correctly and that it is polynomial time bounded in card(N) + length(f). Thus, determination of truth in a R-structure is NP-complete.

(ii) Satisfiability is NP-hard because boolean satisfiability is NP-hard.

Let $f \in L(F)$ and $\mathscr{P}$ = the set of atomic propositions appearing in f. From Theorem 3.1 if f is satisfiable, then it is satisfiable in a structure $S = (s,\xi)$ where size(s) $\le$ length(f). A nondeterministic TM M which checks for satisfiability

of f operates as follows: M guesses init(s) and range(s) such that length(init(s)) ≤ 2 * length(f), card(range(s)) ≤ 2 * length(f). Next it uses a labelling algorithm similar to the one in (i) to accept or reject f. Clearly M is polynomial time bounded in length(f). □

4. The Complexity of L(F,X), L(U) and L(U,S,X)

The main results of this section are summarized in the following theorem.

THEOREM 4.1. *The following problems are* PSPACE-complete *for the logics* L(F,X), L(U), and L(U,S,X):

(i) Satisfiability

(ii) Determination of truth in an R-structure.

The proof of the above theorem is based on the following lemmas.

LEMMA 4.2. *Determining truth in an* R-*structure is polynomial-time reducible to satisfiability for* L(F,X), L(U) *and* L(U,S,X).

Proof: Let $T = (N,R,\eta)$ be an R-structure and let $f \in L(U,S,X)$. Let $\mathcal{P}_1 = \{P_x \mid x \in N\}$ and $\mathcal{P}_1 \cap \mathcal{P} = \emptyset$. $\mathcal{P}_1$ contains one new atomic proposition for each state in N.

$$f_x = G(P_x \supset [\bigwedge_{Q \in \eta(x)} Q \wedge \bigwedge_{Q \in \mathcal{P}-\eta(x)} (\neg Q) \wedge Xg])$$

where g is disjunction of all P_y such that $(x,y) \in R$.

$$f' = (\bigvee_{y \in N} P_y) \wedge (\bigwedge_{x \in N} f_x) \wedge G(\bigwedge_{x \in N} [P_x \supset \bigwedge_{y \neq x} \neg P_y])$$

Any structure $T = (t,\pi)$ such that $T,t_0 \models f'$ has the following property. At each state in t exactly one proposition in $\mathcal{P}_1$ is true, and if P_n is true at a state then all propositions in $\eta(x)$ are true in that state, all propositions in $\mathcal{P} - \eta(x)$ are false in that state, and in the next state P_y is true for exactly one y such that $(x,y) \in R$.

Let $f'' = f' \wedge f \wedge P_q$. It can easily be seen that there is a path p in S starting from q such that $S_p,s_0 \models f$ iff f" is satisfiable. If $f \in L(F,X)$ then $f'' \in L(F,X)$. In f", we can avoid the X operator by using U operator. In this case if $f \in L(U)$ then the resulting formula $\in L(U)$. The above reduction is clearly a polynomial reduction. □

LEMMA 4.3. *Determination of truth in an* R-*structure is* PSPACE-*hard for* L(F,X) *and* L(U).

Proof. Let $M = (Q,\Sigma,\delta,V_A,V_R,V_I)$ be a one tape deterministic TM where Q is the set of states, Σ is the alphabet, $\delta: Q \times \Sigma \to Q \times \Sigma \times \{L,R\}$; V_A,V_R,V_I Q are the accepting, rejecting and initial states respectively. Let M be S(n) space bounded such that S(n) is bounded by a polynomial in n. M halts on all inputs in state V_A or V_R, thus accepting or rejecting the input. An ID of M is appropriately defined. Let $a = a_1a_2\ldots a_n$ be an input to M.

Let $T = (N,R,\eta)$ be an R-structure shown in Figure 2.

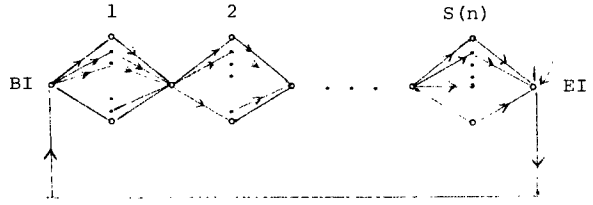

Figure 2

Let $\mathcal{P} = Q \times \Sigma \cup \Sigma \cup \{BI,EI\}$ be the set of atomic propositions. The structure in Figure 2 has S(n) diamonds connected in a chain, and in each diamond there are card(Q x Σ ∪ Σ) number of vertical vertices. In each diamond, on each vertical vertex exactly one atomic proposition is true, and every atomic proposition in Q x Σ ∪ Σ is true on some vertical vertex of the diamond. Each subpath between BI and EI represents one ID of M, and a path from BI represents a sequence of IDs of M.

Using S(n) X operators the relation between the contents of a tapecell in successive IDs can be asserted. Because of this, polynomial length bounded formulas in L(F,X) can be obtained asserting the following conditions: All the IDs on a path p starting from BI are valid, the first ID is the initial ID containing the input string $a_1a_2\ldots a_n$,

each successive ID follows from the previous one by one move of M, and the final ID appears on the path.

Let f_a be the conjunction of formulas asserting the above conditions. It is easily seen that there is a path p from BI in T such that $S_p, s_0 \models f_a$ iff M accepts a. For any input a, f_a can be obtained in polynomial time. We can avoid X operators using only U operator. The resulting formula will be in L(U). □

Let $S = (s,\xi)$ be a structure and $f \in L(U,S,X)$. For any state s_i in s let $[s_i]_{S,f} = \{g \in SF(f) \mid S,s_i \models g\}$.

LEMMA 4.4. *In* $S = (s,\xi)$, *if* s_i, s_j *be two states such that* $[s_i]_{S,f} = [s_j]_{S,f}$ *then for the structure* $S' = (s',\xi')$ *where* $s' = (s_0, s_1, \ldots, s_{i-1}, s_j, s_{j+1}, \ldots)$ *and* ξ' *is restriction of* ξ *to states in* s' *the following property holds:*

For all s_k *such that* s_k *is present in* s *and in* s', $[s_k]_{S,f} = [s_k]_{s',f}$.

The above lemma can be proved by induction on the length of the formula f. □

A structure $S = (s,\xi)$ is said to ultimately periodic with starting index i and period m if $\forall k \geq i \quad \xi(s_k) = \xi(s_{k+m})$. For a structure S and a formula f let $M_{S,f} = \{C \subseteq SF(f) \mid$ there are infinitely many k such that $[s_k]_{S,f} = C\}$.

LEMMA 4.5. *For the structure* $S = (s,\xi)$ *let* i,p *be integers such that* $[s_i]_{S,f} = [s_{i+p}]_{S,f}$ *and for each* $C \in M_{S,f}$ *there is a* k *such that* $i \leq k < i+p$ *and* $[s_k]_{S,f} = C$. *Let* $S' = (s',\xi')$ *be an ultimately periodic structure with starting index* i *and period* p *such that* $\forall k < i+p$ $\xi(s_k) = \xi'(s'_k)$. Then $\forall k < i+p \quad [s_k]_{S,f} = [s'_k]_{S',f}$ *and* $\forall k \geq i \quad [s_k]_{S',f} = [s_{k+p}]_{S',f}$

Proof: By induction it can be proved that for any $g \in SF(f)$,

$\forall k < i+p \quad S,s_k \models g$ iff $S',s'_k \models g$, and

$\forall k \geq i \quad S',s'_k \models g$ iff $S',s'_{k+p} \models g$. □

THEOREM 4.6. *(Ultimately periodic model theorem). A formula* $f \in L(U,S,X)$ *is satisfiable iff it is satisfiable in an ultimately periodic structure* $S = (s,\xi)$ *with starting index* $i \leq 2^{1+length(f)}$, *period* $p \leq 4^{1+length(f)}$ *and* $\forall k \geq i \quad [s_k]_{S,f} = [s_{k+p}]_{S,f}$.

Proof. Let f be satisfiable and $g = Ff$. Then there exists a structure $T \models (t,\eta)$ such that $T,t_0 \models g$. Let ℓ,m be integers such that $[t_\ell]_{T,g} = [t_{\ell+m}]_{T,g}$ and

(*) for each $C \in M_{T,g}$ there exists a state t_k between t_ℓ and $t_{\ell+m}$ such that $[t_k]_{T,g} = C$.

By applying the reduction of Lemma 4.4 repeatedly to states between t_0 and t_ℓ, or to states between t_ℓ and $t_{\ell+m}$ (excluding $t_0, t_\ell, t_{\ell+m}$) while preserving (*), we obtain a structure $T' = (t',\eta')$ such that $T',t'_0 \models g$, and there exist integers $i \leq 2^{length(g)}$, $p \leq 4^{length(g)}$ such that t'_i, t'_{i+p} satisfy the same subformulae and (*) is satisfied for T' with i,p. Using Lemma 4.5 for T' and g we obtain a periodic structure S, with starting index $i \leq 2^{length(g)}$, period $p \leq 4^{length(g)}$ such that $\forall k \geq i \quad [s_k]_{S,g} = [s_{k+p}]_{S,g}$ and $s,s_0 \models g$. □

Proof outline for Theorem 4.1. Let f be a formula in L(U,S,X) and $g = Ff (= True\ Uf)$. f is satisfiable iff g is satisfiable at the beginning of an ultimately periodic structure. We describe below a nondeterministic TM M which checks for satisfiability of g. M guesses two numbers $n_1 \leq 2^{length(g)}$, $n_2 \leq 4^{length(g)}$ which are supposed to be the starting index and period of an ultimately periodic structure. Next, M guesses the subformulae that are true at the beginning, verifies that g is in this set, and checks for consistency.

Subsequently, M guesses the subformulae that are true in the next state and verifies their consistency with the subformulae that are guessed to be true in the present state; e.g., if Xf_1 is present in the guessed subformulae for the present state then f_1 should be in the guessed subformulae for

the next state. It continues this process, each time incrementing the counter. When the counter is n_1, it notes that it is in the periodic part of the structure. It saves the subformulae true at the beginning of the period, and it re-initailizes the counter. It continues guessing the subformulae in the next state and incrementing the counter. At each instance it has to keep three sets of subformulae: those that are true in present state, those true in the next state and those true at the beginning of the period. When the counter has value n_2, it stops guessing and takes the subformulae true at the beginning of the period to be the subfo.mulae true in the next state. At each step in the above procedure it checks the consistency of the subformulae guessed and verifies that certain subformulae are fulfilled; e.g.,if $f_1 \cup f_2$ is in the subformulae guessed to be true at the beginning of the period, then f_2 has to be true somewhere within the period.

5. Complexity of Extensions of the Logic

In [Wo81] propositional linear temporal logic is enriched with the addition of operators corresponding to regular right linear grammars. Let R be a regular right linear grammar with terminal symbols $a_1,a_2,\ldots,a_n$ and non-terminal symbols $N_1,N_2,\ldots,N_m$. If $f_1,f_2,\ldots,f_n$ are formulae in the logic then so is $N_j(f_1,f_2,\ldots,f_n)$ for $1 \le j \le m$. For a structure $S = (s,\xi)$, $S,s_k \models N_j(f_1,f_2,\ldots f_n)$ iff there exists a string $a_{i_1},a_{i_2},a_{i_3}\ldots$ generated by R from N_j such that for all $\ell \ge 0$ $S,s_{k+\ell} \models f_{i_{\ell+1}}$.

Ex: Consider the grammar $N_0 \to a_1a_2N_0$. It generates the infinite string $a_1a_2a_1a_2\ldots$. $S,s_0 \models N_0(\text{True},P)$ iff P holds at all even states in s.

For convenience, we assume that each production rule in the grammar has at most one terminal symbol. For any formula f in this logic we define $SF(f)$ as follows:

If $f = P$ then $SF(f) = \{P\}$;

If $f = f_1 \wedge f_2$ or $f_1 \cup f_2$ or $f_1 S f_2$ then $SF(f) = SF(f_1) \cup SF(f_2) \cup \{f\}$;

If $f = \neg f_1$ or Xf_1 then $SF(f) = SF(f_1) \cup \{f\}$;

If $f = N_j(f_1,f_2,\ldots f_n)$ where N_j is a non-terminal in the above regular grammar, then $SF(f) = \bigcup_{1\le i\le n} SF(f_i) \cup \{N_j(f_1,f_2,\ldots,f_n) \mid 1\le j\le m\}$.

With the above definition of $SF(f)$, Lemma 4.4 and Lemma 4.5 hold for this logic, the proofs being by induction on the length of the formula f. Theorem 4.6 holds if we replace length(f) by card(SF(f)).

In Theorem 4.1, we assume that the grammars corresponding to the regular operators are encoded as part of the input. In this case if the input length is n, then $card(SF(f)) \le n^2$ where f is the input formula. To prove Theorem 4.1, we need to modify the previous proof as follows. The two guessed integers n_1,n_2 should be less than or equal to 2^{n^2}, 4^{n^2} respectively. More consistency checks have to be made whenever a new set of subformulae is guessed, ex: If $N_j(f_1,f_2,\ldots,f_n)$ is in the guessed set, then either there should be a production rule of the form $N_j \to a_kN_\ell$ in the grammar so that f_k is also present in the set of formulae guessed to be true in the the present state and $N_\ell(f_1,f_2,\ldots,f_n)$ is present in the set of formulae guessed to be true in the next state, or there should be a production rule of the form $N_j \to a_k$ so that f_k is also present in the set of subformulae guessed to be true in the present state. It can easily be seen that the TM is polynomial space bounded and accepts any input iff the input formula is satisfiable. □

6. Conclusion

In this paper we have examined the complexity of satisfiability and truth in a particular structure for various propositional linear temporal logics. We have determined that these problems are NP-complete for L(F) and PSPACE-complete for L(F,X), L(U), L(U,S,X), and Wolper's extended logic (see Figure 3). Satisfiability for L(U,X) can also be shown to be in PSPACE by translation into SDPDL [HR81]; however this technique does not work for L(U,S,X) or for Wolper's logic with regular operators. It should be also observed that both the X and G operators are necessary for PSPACE-hardness of satisfiability. Thus, the logic $\tilde{L}(F,X)$ is NP-complete since it does not permit the G operator.

Finally, it is interesting to compare our results with the corresponding results for branching-time logics. Since branching-time formulae are interpreted over the states of a structure rather than over execution sequences, determining truth in a particular structure is much easier and, in many cases, is in P [CE81]. Satisfiability, on the other hand, can be shown to be exponential-time hard for branching-time logics with a nexttime operator and is, therefore, apparently more difficult than in the case of linear-time logics.

Logic	Satisfiability	Validity	Truth in an R-Structure
$L(F)$, $\tilde{L}(F,X)$	NP-complete	CO-NP-complete	NP-complete
$L(F,X)$, $L(U)$, $L(U,X)$, $L(U,S,X)$, Linear time logic with regular operators	PSPACE-complete	PSPACE-complete	PSPACE-complete

$\tilde{L}(F,X)$ is the logic with temporal operators F,X; with boolean connectives $\wedge,\vee$; and with negations allowed only on atomic propositions.

Figure 3

BIBLIOGRAPHY

[BMP81] M. Ben-Ari, Z. Manna, A. Pnueli, "The Temporal Logic of Branching Time." Eighth ACM Symposium on Principles of Programming Languages, Williamsburg, VA, Jan. 1981.

[CE81] Edmund M. Clarke, E. Allen Emerson, "Design and Synthesis of Programming Skeletons Using Branching Time Temporal Logic," IBM Conference on Logics of Programs, May 1981 (to appear in Springer Lecture Notes in Computer Science).

[FL79] M. Fisher, R. Ladner, "Propositional Dynamic Logic of Regular Programs," JCSS, 18(2), 1979.

[GPSS80] D. Gabbay, A. Pnueli, S. Shealah, J. Stavi, "Temporal Analysis of Fairness," Seventh ACM Symposium on Principles of Programming Languages, Las Vegas, NV, Jan. 19 .

[HR81] J. Y. Halpern and J. H. Reif, "The Propositional Dynamic Logic of Deterministic, Well-Structured Programs," 22nd Annual Symposium on Foundations of Computer Science, Nashville, TN, Oct. 1981.

[La77] R. Ladner, "The Computational Complexity of Provability in Systems of Modal Propositional Logic," SIAM J. Comp. 6, 467-480 (A9.Z).

[MP81] Z. Manna, A. Pnueli, "Verification of Concurrent Programs," The Correctness Problem in Computer Science, International Lecture Series in Computer Science, Academic Press, London, 1981.

[Pn77] A. Pnueli, "The Temporal Logic of Programs," Proceedings of the Eighteenth Symposium on Foundations of Computer Science, Providence, RI, Nov. 1977.

[Wo81] P. Wolper, "Temporal Logic Can Be More Expressive," Proceedings of 22nd Symposium on Foundations of Computer Science, Nashville, TN, Oct. 1981.