First Order Linear Temporal Logic over finite temporal structures is not semi-decidable (DRAFT)

Serenella Cerrito[1] and Marta Cialdea Mayer[2] and Sebastien Praud[1]

[1] Université de Paris-Sud, e-mail: {serena,praud}@lri.fr
[2] Università di Roma Tre, e-mail: cialdea@dia.uniroma3.it

1 Introduction

The model of time underlying Linear Temporal Logic (LTL) is a discrete, linear sequence of states, usually taken to be bounded in the past and infinite in the future. In other words, the set of time points is isomorphic to $\mathbb{N}$. In the propositional case, several sound, complete and terminating proof-systems for LTL have been provided; see for instance [6, 5, 3, 2]. In [1] the problem of the existence of a complete recursive axiomatization for the first-order version of LTL containg the following temporal operators: $\Box$ ("always in the future"), $\Diamond$ ("eventually"), $\bigcirc$ ("next"), $\mathcal{U}$ ("until") and $\mathcal{P}$ ("precedes"), equipped with the infinite semantics, is studied. It is proved that such an axiomatization cannot exists. The proof makes use of the notion of Π^1_1 formulae, that is formulae having the form:

$$\forall R_1 \dots \forall R_k \forall F_1 \dots \forall F_{k'} B$$

where B is some classical first-order formula and $0, s, \leq, +, \times, R_1, \dots, R_k, F_1, \dots, F_{k'}$ are all the predicate and function symbols in B. The complexity class Π^1_1 includes all problems no harder than the truth problem for Π^1_1 formulae. What is shown is that the LTL validity problem is Π^1_1-complete. The kernel of the proof consists in exhibiting a recursive embedding E of Π^1_1 formulae in LTL formulae, such that, given any Π^1_1 formula F, F is true if and only if $E(F)$ is LTL valid. The key idea is the following: given a formula F having the form $\forall R_1 \dots \forall R_k \forall F_1 \dots \forall F_{k'} B$, its embedding in LTL is the formula $E(F) =_{def} A \rightarrow B$, where A is a temporal formula having the following property:
for any (temporal) model $\mathcal{M}$ of A, if D is its object-domain and $0_{\mathcal{M}}, s_{\mathcal{M}}, \leq_{\mathcal{M}}, +_{\mathcal{M}}, \times_{\mathcal{M}}$ are the interpretations for $0, s, \leq, +, \times$, then the structure

$$\langle D, 0_{\mathcal{M}}, s_{\mathcal{M}}, \leq_{\mathcal{M}}, +_{\mathcal{M}}, \times_{\mathcal{M}} \rangle$$

is isomorphic to the natural numbers with the standard arithmetic operations.

The definition of the mapping E uses the equality predicate $=$, one non-rigid constant a and only the following temporal operators $\Box$ (always in the future), $\Diamond$ (sometimes in the future), $\bigcirc$ (next in the future). Now, the use of a may be actually simulated by an appropriate use of predicate symbols and, as remarked by Abadi, even the $\bigcirc$ operator can be dispensed with.[1] Hence, the incompleteness theorem remains true for the restricted languages where all function symbols are rigid and the only temporal operators are $\Box$ and $\Diamond$. Let us call first-order $LTL_{\Box,\Diamond}$ this fragment of LTL for which the incompleteness theorem still holds. In this work, we consider a version of first order LTL with a finite semantics, FTL^{fin}, where time is assumed to be finite both in the past and in the future. Its interest is due to the fact that there are problems in Computer Science and Artificial Intelligence where only a finite fragment of the temporal sequence is of interest. For instance, dynamical integrity constraints on databases (evolving through time by means of updates) may be expressed by first order temporal formulae. In particular, the so called DDA constraints [] state that if a given property of data holds at a state of the database history, then another property had been true at all the previous states (for instance: it is necessary to have been a student in order to be a teacher). Thus, at any given time point, only the finite time segment going from the initial time point to the current one is dealt with. Thus FTL^{fin} (or, rather, the quite symetrical logic containing only past-time operators) seems quite apropriate to study dynamical integrity constraints in databases (see [4]).

Obviously, no decision procedure for FTL^{fin} exists, but what about semi-decidability? Now, an essential feature of the formula A used in Abadi's embedding of Π^1_1 into LTL is the capability of A to guarantee that all elements of the domain D represent the natural numbers. This is done by forcing the non rigid constant a to take distinct values in distinct time points, and, for any $d \in D$, to eventually take d as value in some time point. The second condition is expressed by the subformula $\forall x \Diamond a = x$. It is therefore appearent that such a construction cannot be carried over in the case of LTL_{fin} where, at each given time point i, only a bounded future can be "observed".

In this work we show thate the validity problem in FTL_{fin} is not recursively enumerable.

2 FTL^{fin}

The languages of linear temporal logic we consider include the classical operators $\neg, \wedge, \vee$, $\bot$ (always false), $\top$ (always true), $\forall$ and $\exists$ and unary modal operators $\Box$ (always in the future) and $\bigcirc$ (weak next). The alphabet contains a set P of predicate symbols and a set F of function symbols; all the the function symbols in F are *rigid*. The interpretation of the equality predicate is rigid too.

The semantics of the language with predicate symbols in P and function symbols in F is defined as follows. A *temporal frame* T is a finite initial segment of the natural numbers, $\langle 0, \cdots, k \rangle$; its elements are called *time points*. A temporal interpretation $\mathcal{M}$ is a quadruple $\langle T, D, \delta_P, \delta_F \rangle$ where:

- T is a temporal frame,
- D is a non-empty set (the object domain)

[1] Personal communication.

– δ_P is a mapping from $P \times \{0, ..., k\}$ to the set of relations on D such that, for any n-ary predicate symbol p with $n > 0$ and any $i \in T$, $\delta_P(p, i)$ is a subset of D^n;
– δ_F is a mapping from F to the set of (total) functions on D such that, for any n-ary function symbol f with $n \geq 0$, $\delta_F(f)$ is a function from D^n to D.

Note that here we make the *constant domain* hypothesis: the object domain D is the same at each time point.

Let σ be a variable assignment, i.e. a function from the set of variables Var to the domain D. Given any term t, its value $[t]_{\mathcal{M},\sigma}$ w.r.t. the interpretation $\mathcal{M}$ and the variable assignment σ, is defined like in the classical case. Since the language is assumed to have only rigid functional symbols, the interepretion of a term is the same at any time point.

If A is a formula, the satisfiability relation $\mathcal{M}_{\sigma,i} \models A$ (to be read: A is satisfied by $\mathcal{M}$ w.r.t. the variable assignment σ at the time point i) is defined as follows:

1. $\mathcal{M}_{\sigma,i} \models p(t_1, ..., t_n)$ iff $\langle [t_1]_{\mathcal{M},\sigma}, ..., [t_n]_{\mathcal{M},\sigma} \rangle \in \delta_P(p, i)$.
2. $\mathcal{M}_{\sigma,i} \models t_1 = t_2$ iff $[t_1]_{\mathcal{M},\sigma} = [t_n]_{\mathcal{M},\sigma}$.
3. $\mathcal{M}_{\sigma,i} \not\models \perp$.
4. $\mathcal{M}_{\sigma,i} \models \neg A$ iff $\mathcal{M}_{\sigma,i} \not\models A$.
5. $\mathcal{M}_{\sigma,i} \models A \wedge B$ iff $\mathcal{M}_i \models A$ and $\mathcal{M}_{\sigma,i} i \models B$.
6. $\mathcal{M}_{\sigma,i} \models A \vee B$ iff either $\mathcal{M}_{\sigma,i} \models A$ or $\mathcal{M}_{\sigma,i} i \models B$.
7. $\mathcal{M}_{\sigma,i} \models \Box A$ iff for all j such that $i \leq j$, $\mathcal{M}_{\sigma,j} \models A$.
8. $\mathcal{M}_{\sigma,i} \models \Diamond A$ iff there exists $j \in T$ such that $i \leq j$ and $\mathcal{M}_{\sigma,j} \models A$.
9. $\mathcal{M}_{\sigma,i} \models \bigcirc A$ iff either $i + 1 \notin T$ or $\mathcal{M}_{\sigma,i+1} \models A$.

The $\Diamond$ operator ("eventually") is defined as

$$\Diamond A \equiv_{def} \neg\Box\neg A$$

When A is closed, we omit σ and we write just $\mathcal{M}_j \models A$ to mean that A is satisfied by the structure $\mathcal{M}$ at the timepoint j.

Truth of a formula A is the satisfiability of its universal closure in the initial state: a formula A is true in $\mathcal{M}$ (and $\mathcal{M}$ is a model of A) iff $\mathcal{M}_0 \models \forall x_1, ..., \forall x_n A$, where $\{x_1, \cdots, x_n\}$ are all the free variables in A. Truth of sets of formulae is defined as usual. Validity is truth in all models.

3 There exists no r.e. axiomatization for FTL^{fin}

In the rest of this work we prove that FTL^{fin} validity is not recursively enumerable by reducing the complement of the halting problem for Turing Machines:

CO-HALTING PROBLEM
INSTANCE: a Turing machine M and a word w in the alphabet of M.
QUESTION: does M never halts when given w as an input?

to the validity problem in FTL^{fin}.

We consider deterministic Turing Machines running on a tape that is infinite on the left,

and defined by a tuple $\langle Q, \Sigma, \Delta, q_0, q_f \rangle$, where:

- $Q = \{q_0, ..., q_{N_Q}\}$ is a finite set of states,
- $\Sigma = \{s_0, s_1, ..., s_{N_\Sigma}\}$ a finite alphabet, where s_0 is a distinguished symbol (the blank),
- Δ is the transition table of the machine, consisting of a finite set of quadruples of one of the following forms:

  $\langle q_i, s_j, s_k, q_m \rangle$ (if the machine is in state q_i reading symbol s_j, it replaces s_j with s_k and passes in the state q_m)

  $\langle q_i, s_j, L, q_m \rangle$ (if the machine is in state q_i reading symbol s_j, it moves one square to the left and passes in the state q_m)

  $\langle q_i, s_j, R, q_m \rangle$ (if the machine is in state q_i reading symbol s_j, it moves one square to the right and passes in the state q_m)

  For each state $q_i \neq q_f$ and symbol s_j there is at most one transition of the form $\langle q_i, s_j, ... \rangle$ and there is no transition of the form $\langle q_f, ... \rangle$.
- $q_0 \in Q$ is the initial state
- $q_f \in Q$ is the final state

Given a Turing machine M and a word w in the alphabet of M, we define a LTL^{fin} formula $F_{M,w}$ such that $F_{M,w}$ is valid (in all finite temporal structures) iff M never halts when given w as an input.

The language $\mathcal{L}_{M,w}$ of $F_{M,w}$ contains:

1. a constant symbol 0 and a unary functional symbol s; terms of the form $\underbrace{s(s(...(s}_{n}(x))...))$ are abbreviated by $s^{(n)}(x)$.
2. the equality predicate and the binary relation symbol $<$.
3. for each state $q_i \in Q$, the unary predicate symbol Q_i.
4. for each symbol $s_i \in \Sigma$, the unary predicate symbol S_i; we shall use the symbol B for S_0.

We say that M halts before time k if for some time $n < k$, M is in state q_f at time n. We define a set of "intended interpretations" $\mathcal{M}^k$ of the language $\mathcal{L}$, for each $k \geq 0$ such that the machine M does not halt before time k, when given the word w as an input. The time structure in $\mathcal{M}^k$ is $\langle 0, ..., k \rangle$. The domain is $\mathbb{N}$. $\mathcal{M}^k$ assignes zero to 0 and the successor function to s. Moreover, if σ is a variable assignment and $t \in T$:

1. ${\mathcal{M}^k}_{\sigma,t} \models Q_i(x)$ iff at time t the machine is in state q_i, scanning square number $\sigma(x)$
2. ${\mathcal{M}^k}_{\sigma,t} \models S_i(x)$ iff at time t the symbol S_i is in square number $\sigma(x)$
3. ${\mathcal{M}^k}_{\sigma,t} \models x < y$ iff $\sigma(x)$ is less than $\sigma(y)$

3.1 Encoding of the halting problem

Given a turing machine M with input $w = s_{i_0}...s_{i_m}$, the formula $G_{M,w}$ is the conjunction of the following set of formulae:

$$Q_0(0) \wedge S_{i_0}(0) \wedge S_{i_1}(s(0)) \wedge ....S_{i_m}(s^{(m)}(0))$$
$$\wedge \forall x(x \neq 0 \wedge x \neq s(0) \wedge .... \wedge x \neq s^{(m)}(0) \rightarrow B(x)) \tag{G1}$$

$$\Box \forall x \forall y \forall z(s(x) = z \wedge s(y) = z \rightarrow x = y)) \tag{G2}$$

$$\Box(\forall x \forall y \forall z(x < y \wedge y < z \rightarrow x < z)) \tag{G3}$$

$$\Box(\forall x \forall y(s(x) = y \rightarrow x < y)) \tag{G4}$$

$$\Box(\forall x \forall y(x < y \rightarrow x \neq y)) \tag{G5}$$

$$\bigwedge_{i \neq j} \Box \forall x \forall y \neg(Q_i(x) \wedge Q_j(y)) \tag{G6}$$

$$\Box \forall x(Q_f(x) \rightarrow \bigcirc \bot) \tag{G7}$$

For every $\langle q_n, s_j, s_k, q_m \rangle \in \Delta$:

$$\Box \forall x(Q_n(x) \wedge S_j(x) \rightarrow$$
$$(\bigcirc(Q_m(x) \wedge S_k(x))$$
$$\wedge \forall y(x \neq y \rightarrow \bigwedge_{i=0...N_\Sigma}(S_i(y) \rightarrow \bigcirc S_i(y))))) \tag{G8*}$$

For every $\langle q_n, s_j, R, q_k \rangle \in \Delta$:

$$\Box \forall x(Q_n(x) \wedge S_j(x) \rightarrow$$
$$(\bigcirc Q_k(s(x))$$
$$\wedge \forall y(\bigwedge_{i=0...N_\Sigma}(S_i(y) \rightarrow \bigcirc S_i(y))))) \tag{G9*}$$

For every $\langle q_n, s_j, L, q_k \rangle \in \Delta$:

$$\Box \forall x(Q_n(x) \wedge S_j(x) \rightarrow$$
$$(\exists y(s(y) = x \wedge \bigcirc Q_k(y))$$
$$\wedge \forall y(\bigwedge_{i=0...N_\Sigma}(S_i(y) \rightarrow \bigcirc S_i(y))))) \tag{G10*}$$

The first formula, G1, is a description of the initial state. The next three formulae, G2, G3, G4 and G5, state standard properties of s and $<$; from them, the following formula can be derived, for any $n \neq m$:

$$\Box(\forall x(s^{(n)}(x) \neq s^{(m)}(x))) \tag{1}$$

G6 states that the machine can be in at most one state at a time. The fact that time ends when the final state is reached is stated by G7. Finally, the set of formulae of the form G8*, G9* and G10* describe the transition table of the machine.

The formula $H_{M,w}$ is:

$$\Box\forall x \neg Q_f(x) \tag{$H_{M,w}$}$$

intuitively stating that the machine is never in its final state.

Finally, the formula $F_{M,w}$ is $G_{M,w} \rightarrow H_{M,w}$. In the next subsection it is shown that it has the desired properties.

3.2 Reduction of the co-halting problem to the FTL^{fin}-validity problem

A sentence encodes an *instanteous description* of the Turing machine at time t if it has the following form

$$\begin{array}{l}\bigcirc^{(t)}(Q_i(s^{(p)}(0)) \\ \wedge S_{j_1}(s^{(p_1)}(0)) \wedge .... \wedge S_j(s^{(p)}(0)) \wedge .... \wedge S_{j_u}(s^{(p_r)}(0)) \\ \wedge \forall x(x \neq s^{(p_1)}(0) \wedge ... \wedge x \neq s^{(p)}(0) \wedge .... \wedge x \neq s^{(p_r)}(0) \rightarrow B(x)))\end{array}$$

where $\bigcirc^{(t)}$ stands for $\underbrace{\bigcirc ... \bigcirc}_{t}$, i.e. a sequence of t $\bigcirc$, and $p_1, ..., p, ..., p_r$ is an increasing sequence of integers. For example, formula G1 is a description of time 0.

Lemma 1. *For each $t \in \mathbb{N}$, if the machine M has not halted before time t on input $w = s_{i_0}...s_{i_m}$ (i.e. for any $n = 0, ..., t-1$, M is not in the final state), then there exists a description D_t of time t such that $G_{M,w} \models D_t$*

Proof. The proof is by induction on t. If $t = 0$, then the description is G1, and $G_{M,w}$ logically implies G1.

For the inductive step, let us assume that for all $n < t+1$, M is not in the final state q_f at time n. Then the machine does not halt at time t (it is not in the final state) and for all $n < t$, M is not in the final state q_f at time n, so by the inductive hypothesis, $G_{M,w}$ implies some description D_t of time t, of the form:

$$\begin{array}{l}\bigcirc^{(t)}(Q_i(s^{(p)}(0)) \\ \wedge S_{j_1}(s^{(p_1)}(0)) \wedge .... \wedge S_j(s^{(p)}(0)) \wedge .... \wedge S_{j_u}(s^{(p_r)}(0)) \\ \wedge \forall x(x \neq s^{(p_1)}(0) \wedge ... \wedge x \neq s^{(p)}(0) \wedge .... \wedge x \neq s^{(p_r)}(0) \rightarrow B(x)))\end{array} \tag{2}$$

We must show that $G_{M,w}$ implies some description of time $t+1$. Since $\mathcal{M}^{t+1}$ is a model of G, D_t is true in $\mathcal{M}^{t+1}$. Therefore, at time t the machine is in state q_i, scanning square number p on which the symbol s_j occurs. Since the machine does not halt at time t, $q_i \neq q_f$ and there is a transition in Δ of one of the following forms:

1. $\langle q_i, s_j, s_k, q_m \rangle$; in this case $G_{M,w}$ implies the following example of G8*:

$$\begin{aligned}&\Box\forall x(Q_i(x) \wedge S_j(x) \to \\ &\ \bigcirc(Q_m(x) \wedge S_k(x)) \\ &\ \wedge\forall y(x \neq y \to \bigwedge_{i=0\ldots N_\Sigma}(S_i(y) \to \bigcirc S_i(y))))\end{aligned} \tag{3}$$

that implies (since $\Box A \models \bigcirc^{(t} A$ and $\forall x B(x) \models B(s^{(p)})$:

$$\begin{aligned}&\bigcirc^{(t)}\{(Q_i(s^{(p)}(0)) \wedge S_j(s^{(p)}(0)) \to \\ &\ \bigcirc(Q_m(s^{(p)}(0)) \wedge S_k(s^{(p)}(0))) \\ &\ \wedge\forall y(s^{(p)}(0) \neq y \to \bigwedge_{i=0\ldots N_\Sigma}(S_i(y) \to \bigcirc S_i(y)))\}\end{aligned} \tag{4}$$

Since 2 implies

$$\bigcirc^{(t)}(Q_i(s^{(p)}(0)) \wedge S_j(s^{(p)}(0)) \tag{5}$$

it follows (by 4) that $G_{M,w}$ implies

$$\bigcirc^{(t+1)}(Q_m(s^{(p)}(0)) \wedge S_k(s^{(p)}(0)) \tag{6}$$

Moreover, from 4 and 5 it follows, for all S_i, $i = 0, ..., N_\Sigma$:

$$\bigcirc^{(t)}\forall y(s^{(p)}(0) \neq y \to (S_i(y) \to \bigcirc S_i(y))) \tag{7}$$

Now, the following is a consequence of 2

$$\bigcirc^{(t)} S_{j_1}(s^{(p_1)}(0)) \wedge .... \wedge S_j(s^{(p)}(0)) \wedge .... \wedge S_{j_u}(s^{(p_r)}(0)) \tag{8}$$

and for all $j = 1, ..., r$ the following is a consequence of 7:

$$\bigcirc^{(t)} s^{(p)}(0) \neq (s^{(p_j)}(0)) \to (S_i((s^{(p_j)}(0))) \to \bigcirc S_i((s^{(p_j)}(0)))) \tag{9}$$

Since for all $p_j \neq p$, $\Box(s^{(p)}(0) \neq s^{(p_j)}(0))$ is a consequence of 1, hence of $G_{M,w}$, the following is a consequence of $G_{M,w}$ too:

$$\bigcirc^{(t+1)}(S_{j_1}(s^{(p_1)}(0)) \wedge .... \wedge S_k(s^{(p)}(0)) \wedge .... \wedge S_{j_u}(s^{(p_r)}(0))) \tag{10}$$

From 2, it follows:

$$\bigcirc^{(t)}\forall x(x \neq s^{(p_1)}(0) \wedge x \neq s^{(p)}(0) \wedge .... \wedge x \neq s^{(p_r)}(0) \to B(x))) \tag{11}$$

and this, with 7, implies

$$\bigcirc^{(t+1)}\forall x(x \neq s^{(p_1)}(0) \wedge x \neq s^{(p)}(0) \wedge .... \wedge x \neq s^{(p_r)}(0) \to B(x)) \quad (12)$$

Putting 6, 10 and 12 together, $G_{M,w}$ logically implies:

$$\begin{aligned}&\bigcirc^{(t+1)}(Q_m(s^{(p)}(0)) \\ &\wedge S_{j_1}(s^{(p_1)}(0)) \wedge .... \wedge S_k(s^{(p)}(0)) \wedge .... \wedge S_{j_u}(s^{(p_r)}(0)) \\ &\wedge \forall x(x \neq s^{(p_1)}(0) \wedge x \neq s^{(p)}(0) \wedge .... \wedge x \neq s^{(p_r)}(0) \to B(x)))\end{aligned} \quad (13)$$

that is a description of state $t+1$.

2. $\langle q_i, s_j, R, q_m\rangle$; in this case $G_{M,w}$ implies the following example of G9*:

$$\begin{aligned}&\Box\forall x(Q_i(x) \wedge S_j(x) \to \\ &(\bigcirc Q_m(s(x)) \\ &\wedge \forall y(\bigwedge_{i=0\ldots N_\Sigma}(S_i(y) \to \bigcirc S_i(y)))))\end{aligned} \quad (14)$$

hence also:

$$\begin{aligned}&\bigcirc^{(t)}(Q_i(s^{(p)}(0)) \wedge S_j(s^{(p)}(0)) \to \\ &(\bigcirc Q_m(s(s^{(p)}(0))) \\ &\wedge \forall y(\bigwedge_{i=0\ldots N_\Sigma}(S_i(y) \to \bigcirc S_i(y)))))\end{aligned} \quad (15)$$

Since 2 implies:

$$\bigcirc^{(t)}(Q_i(s^{(p)}(0)) \wedge S_j(s^{(p)}(0)) \quad (16)$$

it follows (by 15) that $G_{M,w}$ implies

$$\bigcirc^{(t+1)}Q_m(s^{(p+1)}(0)) \quad (17)$$

A similar reasoning as in the above case yelds to prove that $G_{M,w}$ implies:

$$\begin{aligned}&\bigcirc^{(t+1)}S_{j_1}(s^{(p_1)}(0)) \wedge .... \wedge S_j(s^{(p)}(0)) \wedge .... \wedge S_{j_u}(s^{(p_r)}(0)) \\ &\wedge \forall x(x \neq s^{(p_1)}(0) \wedge x \neq s^{(p)}(0) \wedge .... \wedge x \neq s^{(p_r)}(0) \to B(x)))\end{aligned} \quad (18)$$

hence, it implies

$$\begin{aligned}&\bigcirc^{(t+1)}(Q_m(s^{(p+1)}(0)) \\ &\wedge S_{j_1}(s^{(p_1)}(0)) \wedge .... \wedge S_j(s^{(p)}(0)) \wedge .... \wedge S_{j_u}(s^{(p_r)}(0)) \\ &\wedge \forall x(x \neq s^{(p_1)}(0) \wedge x \neq s^{(p)}(0) \wedge .... \wedge x \neq s^{(p_r)}(0) \to B(x)))\end{aligned} \quad (19)$$

that is a description of state $t+1$.

3. $\langle q_i, s_j, L, q_m\rangle$; in this case $G_{M,w}$ implies the following example of G10*:

$$\begin{aligned}&\Box\forall x(Q_i(x) \wedge S_j(x) \rightarrow \\ &\ (\exists y(s(y) = x \wedge \bigcirc Q_m(y)) \\ &\ \wedge \forall y(\bigwedge_{i=0\ldots N_\Sigma} (S_i(y) \rightarrow \bigcirc S_i(y)))))\end{aligned} \tag{20}$$

hence also:

$$\begin{aligned}&\bigcirc^{(t)}(Q_i(s^{(p)}(0)) \wedge S_j(s^{(p)}(0)) \rightarrow \\ &\ (\exists y(s(y) = s^{(p)}(0) \wedge \bigcirc Q_m(y)) \\ &\ \wedge \forall y(\bigwedge_{i=0\ldots N_\Sigma} (S_i(y) \rightarrow \bigcirc S_i(y)))))\end{aligned} \tag{21}$$

Since 2 implies:

$$\bigcirc^{(t)}(Q_i(s^{(p)}(0)) \wedge S_j(s^{(p)}(0)) \tag{22}$$

it follows that $G_{M,w}$ implies

$$\bigcirc^{(t)}(\exists y(s(y) = s^{(p)}(0) \wedge \bigcirc Q_m(y))) \tag{23}$$

Now, since the machine can move one square to the left, $p > 0$, and $s(s^{(p-1)}(0)) = s^{(p)}(0)$. As an instance of G2, we have:

$$\Box\forall y(s(s^{(p-1)}(0)) = s^{(p)}(0) \wedge s(y) = s^{(p)}(0) \rightarrow s^{(p-1)}(0) = y)) \tag{24}$$

hence also:

$$\bigcirc^{(t)}\forall y(s(s^{(p-1)}(0)) = s^{(p)}(0) \wedge s(y) = s^{(p)}(0) \rightarrow s^{(p-1)}(0) = y)) \tag{25}$$

so that $G_{M,w}$ implies

$$\bigcirc^{(t)}(s(s^{(p-1)}(0)) = s^{(p)}(0) \rightarrow \bigcirc Q_m(s^{(p-1)}(0))) \tag{26}$$

and also:

$$\bigcirc^{(t+1)}Q_m(s^{(p-1)}(0))) \tag{27}$$

Again we can prove that $G_{M,w}$ implies:

$$\begin{aligned}&\bigcirc^{(t+1)}S_{j_1}(s^{(p_1)}(0)) \wedge .... \wedge S_j(s^{(p)}(0)) \wedge .... \wedge S_{j_u}(s^{(p_r)}(0)) \\ &\ \wedge \forall x(x \neq s^{(p_1)}(0) \wedge x \neq s^{(p)}(0) \wedge .... \wedge x \neq s^{(p_r)}(0) \rightarrow B(x)))\end{aligned} \tag{28}$$

hence, it implies

$$\begin{aligned}&\bigcirc^{(t+1)}(Q_m(s^{(p-1)}(0)) \\ &\ \wedge S_{j_1}(s^{(p_1)}(0)) \wedge .... \wedge S_j(s^{(p)}(0)) \wedge .... \wedge S_{j_u}(s^{(p_r)}(0)) \\ &\ \wedge \forall x(x \neq s^{(p_1)}(0) \wedge x \neq s^{(p)}(0) \wedge .... \wedge x \neq s^{(p_r)}(0) \rightarrow B(x)))\end{aligned} \tag{29}$$

that is a description of state $t + 1$.

Theorem 1. *For any turing machine M with input w:*

$$\models G \rightarrow H \text{ iff } M \text{ never halts when given } w \text{ as an input.}$$

Proof.

1. (*Only if* part). Let us assume that $\models G_{M,w} \rightarrow H_{M,w}$. $G_{M,w}$ is true in all the intended interpretations $\mathcal{M}^k$, hence for all of them $\mathcal{M}^k \models H_{M,w}$.
   Now, let us assume that M halts at time t when given w as an input, i.e. M is in the final state q_f at time t. Since M does not halt before time t, there exists the intended interpretation $\mathcal{M}^t$ of $\mathcal{L}_{M,w}$ of size t, and for some variable assignment σ, $\mathcal{M}^t_{\sigma,t} \models Q_f(x)$. But, since $\mathcal{M}^t_0 \models \Box\forall x \neg Q_f(x)$, $\mathcal{M}^t_{\sigma,t} \not\models Q_f(x)$: contradiction. Therefore, M never halts.
2. (*If* part). Hypothesis: M never halts when given w as an input. Let us assume that $\not\models G_{M,w} \rightarrow H_{M,w}$, i.e. $G_{M,w} \wedge \neg H_{M,w}$ is satisfiable. Let $\mathcal{M}$ be a model of $G_{M,w} \wedge \neg H_{M,w}$ and let k be its last time point. Since $\mathcal{M} \models \neg\Box\forall x \neg Q_f(x)$, there is a time point $fin \leq k$ such that

   $$\mathcal{M}_{fin} \not\models \forall x \neg Q_f(x)$$

   i.e. there is a variable assigment σ such that

   $$\mathcal{M}_{fin,\sigma} \models Q_f(x)$$

   Since $\mathcal{M}$ is a model of G7:

   $$\mathcal{M}_0 \models \Box\forall x(Q_f(x) \rightarrow \bigcirc\bot)$$

   we get $fin = k$.
   Under the hypothesis that M never halts on w, by Lemma 1, $G_{M,w}$ logically implies some description D_k of time k, of the form:

   $$\bigcirc^{(k)}(Q_i(s^{(p)}(0)) \wedge S_{j_1}(s^{(p_1)}(0)) \wedge .... \wedge S_j(s^{(p)}(0)) \wedge .... \wedge S_{j_u}(s^{(p_r)}(0))$$
   $$\wedge \forall x(x \neq s^{(p_1)}(0) \wedge x \neq s^{(p)}(0) \wedge .... \wedge x \neq s^{(p_r)}(0) \rightarrow B(x)))$$

   Therefore, $\bigcirc^{(k)}Q_i(s^{(p)}(0))$ is a logical consequence of $G_{M,w}$, for some i and p.
   Now, if it were $i \neq f$, than also

   $$\Box\forall x\forall y\neg(Q_i(x) \wedge Q_f(x))$$

   would be a consequence of $G_{M,w}$ (from G6), so that

   $$G_{M,w} \models \Box(Q_i(s^{(p)}(0)) \rightarrow \forall y \neg Q_f(y))$$

   contradicting the fact that $\mathcal{M}_{fin,\sigma} \models Q_f(x)$. Hence $i = f$, so that $G_{M,w}$ logically implies

   $$\bigcirc^{(k)}Q_f(s^{(p)}(0))$$

   Therefore, also $\mathcal{M}^k \models \bigcirc^{(k)}Q_f(s^{(p)}(0))$, where $\mathcal{M}^k$ is the intended interpretation of size k. It follows that at time k the machine M, when given w as an input, is in state q_f scanning square number p. I.e. M halts on w: contradiction.

As a consequence of the existence of a reduction of the co-halting problem, which is not recursively enumerable, to the LTL^{fin} validity problem, we get the following result:

Theorem 2. *The LTL^{fin} validity problem is not recursively enumerable.*

Remark. The reason why the proof of the above theorem could not be adapted to the usual encodings of Turing machines in classical logic is that, in the proof of the *if* part, the time point fin could be a non standard number.

4 Conclusions

Some similarity between Theorem 2 and Trakhtentot's theorem, stating the set of classical first order formulae which are true in all the interpretations having a finite domain may be observed. Note, however, that the complement of this last problem is obviously recursively enumerable, so that to prove Trakhtentot's theorem it suffices to code by a classical first order logic formula $F_{M,w}$ a Turing machine M in such a way that M halts on input w if and only if $F_{M,w}$ is satisfiable on a finite domain. In the case of LTL^{fin}, neither validity nor satisfiability are recursively enumerable.

As a consequence of Theorem 2, we get that no effective proof system for LTL^{fin} may exist (provided that both $\Box$ and $\bigcirc$ are there.)

References

1. M. Abadi. The power of temporal proofs. *Theoretical Computer Science*, 65:35–83, 1989.
2. M. Abadi and Z. Manna. Nonclausal deduction in first-order temporal logic. *Journal of the Association for Computing Machinery*, 37:279–317, 1990.
3. H. Barringer, M. Fisher, D. Gabbay, G. Gough, and R. Owens. METATEM: a framework for programming in temporal logic. In *Proc. of REX Workshop on Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness*, volume 430 of *LNCS*. Springer, 1989.
4. V. Benzaken, S. Cerrito, and Praud S. Static preservation of dynamic constraints: a semantics-based approach (extended abstract). Submitted.
5. P.H. Schmitt and J. Goubault-Larrecq. A tableau system for linear-time temporal logic. In E. Brinksma, editor, *3rd Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'97)*, LNCS. Springer Verlag, 1997.
6. P. Wolper. The tableau method for temporal logic: an overview. *Logique et Analyse*, 28:119–152, 1985.