A Tabelau Calculus for First Order Linear Temporal Logic over Bounded Time Structures (DRAFT)

Serenella Cerrito[1] and Marta Cialdea Mayer[2]* and Sébastien Praud[1]

[1] Université de Paris-Sud, L.R.I., 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 $\mathbb{N}$. Different sets of temporal operators may be considered: mainly, future time operators ($\Box$: always, $\Diamond$: eventually, $\bigcirc$: next, $\mathcal{U}$: until), possibly restricting to the fragment with $\Box$ and $\Diamond$ only, or both past and future time ones ("full" LTL).

In the propositional case, several sound, complete and terminating proof-systems for LTL have been provided; see for instance [11, 10, 3, 2]. Not much work has been done on the first-order case. In [?] a normal form theorem for first-order LTL formulae is proved, but no proof-system is provided. As far as first order LTL, equipped with the infinite semantics, is concerned, in [1] it is proved that no complete recursive axiomatization can exist for the logic containg the temporal operators $\Box$ ("always in the future"), $\Diamond$ ("eventually"), $\bigcirc$ ("next"), $\mathcal{U}$ ("until") and $\mathcal{P}$ ("precedes").

In this work, we consider versions of first order LTL with a finite semantics, $FO\text{-}LTL^{fin}$, where time is assumed to be finite both in the past and in the future, while the object domain can be infinite. 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 *Anteriority Dynamic Dependencies* [5] state that if a given property of data holds at a state of the database history, then another property had been true at 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, corresponding to database creation,up to the current one is dealt with. Indeed, finite temporal structures appear to be quite appropriate to study dynamical integrity constraints in databases [4].

* This work has been partially supported by MURST, ASI and CNR.

Similarly, the specification of a planning problem may be expressed by means of a set of temporal logic formulae; since a plan is a finite sequence of actions, leading from the initial situation to the desired goal, it can be modeled by a temporal model over a finite time structure [7, 8].

Let us consider now any LTL fragment containing only past time operators, this time equipped with the *infinite* semantics. Intuitively, although time is unbounded in the future, satisfiability of a formula F of such a logic at a time point i of some interpretation can only express which properties hold in the finite segment of time that goes from i backward to the initial time point 0. Therefore, the satisfiability problem for such a logic is quite analogous to the satisfiability problem for $FO\text{-}LTL^{fin}$, and an answer with respect to the possibility or recursively axiomatizing $FO\text{-}LTL^{fin}$ may also provide insights on the possibility of axiomatizing LTL with the infinite semantics but only past time operators.

Obviously, no decision procedure for $FO\text{-}LTL^{fin}$ exists, but what about semi-decidability? Abadi's negative result about first order LTL over infinite time structures cannot be adapted to the case of $FO\text{-}LTL^{fin}$, but in [6] it is shown that the validity problem for $FO\text{-}LTL^{fin}$ is not recursively enumerable. As a consequence, no effective proof system for $FO\text{-}LTL^{fin}$ may exist.

However, in the case of LTL over finite time structures, a weaker notion than validity can be considered. If k is any given positive inteer, the notion of k-validity of a formula is defined as follows: F is k-valid iff it is true in any finite interpretation whose underlying time structure is not greater than k. The notion of bounded validity is still a useful one in several applications, for instance, in databases, where the question of the preservation of an integrity constraint after a given number k of updates is important, and in planning, where the searched plan can be reasonably required not to exceed a given length.

In this work we show that the notion of *bounded* $FO\text{-}LTL^{fin}$-validity is semidecidable, by exhibiting a sound and complete proof system for this logic. The system is based on analytical tableaux and is an extension of the propositional calculus presented in [7]. When no upper bound on the size of temporal frames is given as an input, the system is sound but obvioulsy incomplete w.r.t. standard $FO\text{-}LTL^{fin}$ validity, but, when such an input k is provided, the system is sound and complete w.r.t. k-validity.

2 Syntax and Semantics of $FO\text{-}LTL^{fin}$

The language of linear temporal logic we consider includes the classical operators $\neg, \wedge, \vee$, $\bot$ (always false), $\top$ (always true), $\forall$ and $\exists$, the unary modal operators $\Box$ (always in the future), $\Diamond$ (eventually), $\overleftarrow{\Box}$ (always in the past), $\overleftarrow{\Diamond}$ (sometime in the past), and the binary ones $\mathcal{U}$ (until) and $\mathcal{S}$ (since). 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*. [1] When the distinguished equality predicate $=$

[1] Note that non-rigid function symbols may always be simulated via predicates. Thus, considering only rigid function symbols does not affect the expressive power of the considered logic.

is considered, it is also taken to be rigid, in the sense that its interpretation does not change over time.

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, possibly infinite)
- 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. Note that, since the language is assumed to have only rigid functional symbols, the interpretation 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) extends the classical definition 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. $\mathcal{M}_{\sigma,i} \models \Box A$ iff for all j such that $i < j$, $\mathcal{M}_{\sigma,j} \models A$.
3. $\mathcal{M}_{\sigma,i} \models \Diamond A$ iff there exists $j \in T$ such that $i < j$ and $\mathcal{M}_{\sigma,j} \models A$.
4. $\mathcal{M}_{\sigma,i} \models A\mathcal{U}B$ iff $\exists j \in T$ such that $i < j$ and $\mathcal{M}_{\sigma,j} \models B$, and for any k with $i < k < j$ $\mathcal{M}_{\sigma,k} \models A$.
5. $\mathcal{M}_{\sigma,i} \models \overleftarrow{\Box} A$ iff for all $j < i$, $\mathcal{M}_{\sigma,j} \models A$.
6. $\mathcal{M}_{\sigma,i} \models \overleftarrow{\Diamond} A$ iff there exists $j \in T$ such that $j < i$ and $\mathcal{M}_{\sigma,j} \models A$.
7. $\mathcal{M}_{\sigma,i} \models A\mathcal{S}B$ iff $\exists j \in T$ such that $j < i$ and $\mathcal{M}_{\sigma,j} \models B$ and for any k with $j < k < i$ $\mathcal{M}_{\sigma,k} \models A$.

Note that, due to the "strong" interpretation of the modal operators (excluding the present time point in both cases of future and past time operators), the weak and strong "Next" and "Last" operators are definable. Although the $\Diamond$ and $\overleftarrow{\Diamond}$ operators are also definable in terms of, respectively, $\mathcal{U}$ and $\mathcal{S}$, we prefer to give them a separate treatment in the tableau system. The other operators taken as primitive are all necessary because the tableau calculus treats formulae in negation normal form.

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

Truth of a formula A is 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. Satifiability of a closed formula is truth in at least a model, validity is truth in all models.[2] If k is a positive integer and A a closed formula, A is said to be k-satisfiable if it is true in some model whose temporal frame $\langle 0, ..., m\rangle$ is such that $m \leq k$. A is k-valid iff it is true in any model whose temporal frame $\langle 0, ..., m\rangle$ is such that $m \leq k$.

A formula is in *negation normal form* (nnf) iff no logical operator is in the scope of a negation. Two formulae A and B are equivalent iff for all $\mathcal{M}$ and i, $\mathcal{M}_i \models A$ iff $\mathcal{M}_i \models B$. It can easily be shown that, under this strong notion of equivalence, every formula can be transformed into an equivalent formula in nnf, by applying the usual classical equivalences and:

$$\begin{array}{ll} \neg\Box A \equiv \top\mathcal{U}\neg A & \neg(A\mathcal{U}B) \equiv (\neg B\mathcal{U}(\neg A \wedge \neg B)) \vee \Box\neg B \\ \neg\overleftarrow{\Box} A \equiv \top\mathcal{S}\neg A & \neg(A\mathcal{S}B) \equiv (\neg B\mathcal{S}(\neg A \wedge \neg B)) \vee \overleftarrow{\Box}\neg B \end{array}$$

3 First Order Bounded Tableaux

In this section, we propose a labelled tableau system to check k-satisfiability of $FO\text{-}LTL^{fin}$ formulae. It extends the system introduced in [7] for the propositional case, whose purpose is to verify whether a set of formulae in nnf is finitely satisfiable (the restriction to nnf formulae is introduced only to simplify the presentation of the rules). The main feature of the system, that is inspired by [9] and [10], is that of explicitely denoting time points and having tableau nodes labelled by either expressions intuitively stating that a formula holds in a given temporal interval, or "temporal constraints", i.e. linear disequations on pairs of time points. Branch closure is reduced to unsatisfiability over the integers of the set of temporal constraints in the branch.

Let $C = \{start, finish, d_1, d_2, d_3, ...\}$ be a set of *time constants* (intuitively denoting time points, where $start$ is the initial time point and $finish$ the last one). A *state* is any expression of the form $c + n$, for $c \in C$ and $n \in \mathbb{Z}$. The set of states is denoted by Σ. It is intended that $C \subset \Sigma$ (c can be rewritten as $c + 0$). If $s, t \in \Sigma$, then $s \leq t$ is a *temporal constraint*. A *labelled formula* is an expression of the form $[s, t]A$, where $s, t \in \Sigma$ and A is a closed LTL formula in nnf (intuitively meaning that A holds at each point in the closed interval $[s, t]$). $[s, s]A$ will be abbreviated by $[s]A$.

Tableau nodes are labelled either by temporal constraints or labelled formulae (in this last case they are called *logical nodes*). If S is a finite set of formulae in nnf and K is the singleton constraint set $\{finish \leq start + k\}$ for some integer $k \geq 0$ (representing the maximal size of the searched models), then tableaux for $S \cup K$ are initialized with the set $\{[start]A \mid A \in S\} \cup K$. Nodes are expanded by application of the rules in Table 1 (the rules for $\Diamond$ and $\overleftarrow{\Diamond}$ are omitted here, they can be found in [7]), where the expressions $s, t, s', t', ..$ are elements of Σ, C is a set of time constants and $P = \{a_1, a_2, a_3, ...\}$ is a set of "fresh" parameters

[2] A different definition of validity can also be found in the literature: A is valid iff for any temporal interpretation $\mathcal{M}$ and for any time point i, $\mathcal{M}_i \models A$. However, it is easy to see that the two notions of validity are interdefinable.

(whose role is to "give names" to elements of the object domain), treated as individual constants. The set of nodes occurring above the line of a rule is called the *premise* of the rule, while the sets of nodes occurring below are the *expansions* of the premise.

In the δ-rule, c is a fresh element of C and a is a fresh element of P. In the γ-rule, u is any ground term occurring in the branch. Note that the γ-rule contains a contraction, in order to make it explicit the fact that a universal formula may be used several times. Moreover, a sort of contraction is implicit in the β-rule and in the δ-rule: in both cases, the rightmost expansion of the rule contains a node with the same formula already occurring in the premise, even though the labels (intervals) of the nodes are different.

Logical rules			
α-rule	**β-rule**	**$\mathcal{U}$-rule**	**$\Box$-rule**
$\dfrac{\begin{array}{c}[s,t]\,A\wedge B\\ s\le t\end{array}}{\begin{array}{c}[s,t]\,A\\ [s,t]\,B\end{array}}$	$\dfrac{\begin{array}{c}[s,t]\,A\vee B\\ s\le t\end{array}}{[s,t]\,A \qquad \begin{array}{c}s\le c\\ c\le t\\ [s,c-1]\,A\\ [c]\,B\\ [c+1,t]\,A\vee B\\ c\in C \text{ fresh}\end{array}}$	$\dfrac{\begin{array}{c}[s,t]\,A\mathcal{U}B\\ s\le t\end{array}}{\begin{array}{c}[c]\,B\\ t+1\le c\\ [t+1,c-1]\,A\\ [s+1,t]\,A\vee B\\ c\in C\ fresh\end{array}}$	$\dfrac{\begin{array}{c}[s,t]\,\Box A\\ s\le t\end{array}}{[s+1,finish]\,A}$
γ-rule	**δ-rule**	**$\mathcal{S}$-rule**	**$\overleftarrow{\Box}$-rule**
$\dfrac{\begin{array}{c}[s,t]\,\forall x A(x)\\ s\le t\end{array}}{\begin{array}{c}[s,t]\,A(u)\\ [s,t]\,\forall x A(x)\end{array}}$	$\dfrac{\begin{array}{c}[s,t]\,\exists x A(x)\\ s\le t\end{array}}{[s,t]\,A(a) \qquad \begin{array}{c}s\le c\\ c\le t\\ [s,c]\,A(a)\\ [c+1,t]\,\exists x A(x)\\ c\in C\ fresh\\ a\in P\ fresh\end{array}}$	$\dfrac{\begin{array}{c}[s,t]\,A\mathcal{S}B\\ s\le t\end{array}}{\begin{array}{c}[c]\,B\\ c\le s-1\\ [c+1,s-1]\,A\\ [s,t-1]\,A\vee B\\ c\in C\ fresh\end{array}}$	$\dfrac{\begin{array}{c}[s,t]\,\overleftarrow{\Box}A\\ s\le t\end{array}}{[start,t-1]\,A}$

Interval rule	**Conflict resolution rules**	
$\dfrac{[s,t]\,A}{t\le s-1 \qquad s\le t}$	$\dfrac{\begin{array}{c}s\le t\\ s'\le t'\\ [s,t]\,p\\ [s',t']\,\neg p\end{array}}{t\le s'-1 \qquad t'\le s-1}$	$\dfrac{[s,t]\,\bot}{t\le s-1}$

Table 1. Tableau expansion rules

The intuition behind the β-rule is the following: either A is true in the whole interval (leftmost branch), or there exists a smallest time point c in the interval where A is false, hence B is true; since c is chosen to be the first of such points,

A is true in the (possibly empty) subinterval before it. Similarly, the δ-rules can be read as stating that if $\exists x A(x)$ holds in the non empty interval $[s,t]$, then either $A(a)$ holds at each point of the interval for the same object a (leftmost branch), or $A(a)$ holds for the same object A in a first segment $[s,c]$ of the interval (possibly with $c = s$), and in the remaining part $\exists x A(x)$ holds again.

The interval and conflict resolution rules augments the set of temporal constraints in the tableau. When the interval rule is applied to expand $[s,t]A$, we say that it is applied to the interval $[s,t]$, independently of the formula A. This rule distinguishes the cases where an interval is empty or not. Its role is to provide the preconditions for the application of the logical and resolution rules. Intuitively, it is useless – and sometimes incorrect – to expand a node $[s,t]A$ when the interval is empty and, given two nodes $[s,t]p$, $[s',t']\neg p$, there is no conflict to be solved if either $[s,t]$ or $[s',t']$ (or both) are empty. Note that such a rule could be dispensed with, and a corresponding branching added to most of the other rules, handling the case where the considered interval is empty.

When the leftmost conflict resolution rule is applied, we say that it is applied to the nodes $[s,t]\,p$ and $[s',t']\,\neg p$.
Standard rules for the equality predicate $=$ on ground terms may be added, so as to deal with equality.

In the following, if $\mathcal{B}$ is a tableau branch, $const(\mathcal{B})$ denotes the set of temporal constants (elements of C) occurring in $\mathcal{B}$ and including $start$ and $finish$.

Definition 1. *Let C be a set of constants (including start and $finish$) and $\mathcal{I}$ a mapping from C to the natural numbers. The notation $\mathcal{I}^*$ is used to denote the extension of $\mathcal{I}$ from states to the integers such that $\mathcal{I}^*(c+n) = \mathcal{I}(c) + n$ for every $c \in C, n \in \mathbb{Z}$.*

1. *Let $T = \langle 0, ..., k\rangle$ be a finite sequence of integers starting at 0. $\mathcal{I}$ is a* temporal mapping *for C with range T if $min\{\mathcal{I}(c) \mid c \in C\} = \mathcal{I}(start) = 0$ and $max\{\mathcal{I}(c) \mid c \in C\} = \mathcal{I}(finish) = k$. Hence, in particular, the range of a temporal mapping is always finite.*
2. *If K is a set of temporal constraints over C, then $\mathcal{I}$ is a solution to K iff:*
   (a) $\mathcal{I}$ is a temporal mapping for C;
   (b) if $s \leq t \in K$, then $\mathcal{I}^*(s) \leq \mathcal{I}^*(t)$.
3. *Let $\mathcal{B}$ be a tableau branch, $C = const(\mathcal{B})$ and $\mathcal{M}$ a temporal interpretation with domain T.*
   (a) If $\mathcal{I}$ is a temporal mapping for C with range T, then $\langle \mathcal{M}, \mathcal{I}\rangle$ satisfies $\mathcal{B}$ ($\langle \mathcal{M}, \mathcal{I}\rangle \models \mathcal{B}$) iff:
      i. $\mathcal{I}$ is a solution to the set of temporal constraints occurring in $\mathcal{B}$;
      ii. if $[s,t]A$ occurs in $\mathcal{B}$, then for every integer i, if $i \in T$ and $\mathcal{I}^*(s) \leq i \leq \mathcal{I}^*(t)$, then $\mathcal{M}_i \models A$.
   (b) $\mathcal{B}$ is satisfiable *in $\mathcal{M}$ iff there exists a temporal mapping $\mathcal{I}$ for C such that $\langle \mathcal{M}, \mathcal{I}\rangle \models \mathcal{B}$.*

Definition 2. *Let $\mathcal{B}$ be a tableau branch and K the set of temporal constraints occurring in $\mathcal{B}$. $\mathcal{B}$ is* open *iff there exists a solution to K. Otherwise it is* closed.

The following definition captures the intuitive idea of tableaux where no wasteful expansions are ever performed. In particular, closed branches are never expanded.

Definition 3. *A tableau branch $\mathcal{B}$ is canonical iff:*

- *The interval rule is applied no more than once to each interval.*
- *Every labelled node is expanded no more than once by means of a logical rule.*
- *If two nodes having an identical label $[s,t]\forall x A$ are expanded to, respectively, $[s,t]A(u)$ and $[s,t]A(u')$ via a γ-rule, then the ground terms u and u' are syntactically different.*
- *The conflict resolution rule is applied no more than once to each node or pair of nodes.*
- *No proper initial subsegment of $\mathcal{B}$ is closed.*

A tableau is canonical iff all its branches are canonical.

Definition 4. *If $\mathcal{B}$ is a tableau branch, then $\mathcal{B}$ is* complete *iff there exists no canonical expansion of $\mathcal{B}$. A tableau is complete if all its branches are complete.*

Obviously, like in classical tableau system, a canonical open branch may be infinite, because of the contraction in the γ rule. More important, a priori, nothing ensures that a canonical *closed* branch is finite. In fact, an infinite branch may be closed (i.e. unsatisfiable) because its infinite set K of constraints is unsatisfiable, yet any finite subset of K has a (finite) solution. As an example, consider $K = \bigcup_{i \in \mathbb{N}} \{c_i \leq c_{i+1} - 1\}$: each finite subset of K has a solution, however K is not finitely satisfiable. Hence, in principle, a canonical tableau might be closed (all its branches being closed) and yet infinite (at least one branch being infinite). However, in the next section, we show that, *thanks to the presence of the initial constraint $finish \leq start + k$* (setting a bound on the size of the searched models), closed tableaux are always finite.

4 Properties of the Bounded Tableau Calculus

Let $a_1, ..., a_n$ be elements of the set of parameters P and let $\mathcal{M}$ be a temporal interpretation that *does not* interpret $a_1, .., a_n$. In the sequel, we use the expression "extension of $\mathcal{M}$ to $a_1, ..., a_n$" to denote any model $\mathcal{M}'$ that is defined exactly like $\mathcal{M}$, for the language interpreted by $\mathcal{M}$, and that assignes values also to the new parameters..

Lemma 1. *Let $\mathcal{B}$ be a branch in a tableau $\mathcal{T}$, $\mathcal{M}$ a temporal interpretation and $\mathcal{I}$ a temporal mapping for the constants occurring in $\mathcal{B}$. Suppose that $\langle \mathcal{M}, \mathcal{I} \rangle \models \mathcal{B}$ and $\mathcal{B}$ can be expanded, $\mathcal{B}_1$ (and $\mathcal{B}_2$) being its expansion(s). Then for some $i = 1, 2$ there exists an extension $\mathcal{I}'$ of $\mathcal{I}$ to the fresh constants of $\mathcal{B}_i$ (if any) and an extension $\mathcal{M}$' of $\mathcal{M}$ to the new parameters in $\mathcal{B}_i$ (if any) such that $\langle \mathcal{M}', \mathcal{I}' \rangle \models \mathcal{B}_i$*

Proof. Let $[s,t]F$ be any labeled formula in $\mathcal{B}$, to which an expansion rule has been applied. It suffices to prove that if $\mathcal{I}^*(s) \leq \mathcal{I}^*(t)$ and, for any time point i in $[\mathcal{I}^*(s), \mathcal{I}^*(t)]$, $\mathcal{M}_i \models F$, then there is at least one expansion $\mathcal{B}_i$ of $[s,t]F$ such that there are an extension $\mathcal{I}'$ of $\mathcal{I}$ to the fresh temporal constants of $\mathcal{B}_i$ (if any) and an extension $\mathcal{M}$' of $\mathcal{M}$ to the new parameter in $\mathcal{B}_i$ (if any) enjoying the following properties:

1. for any labeled formula $[s',t']F'$ in $\mathcal{B}_i$, for any time point i in $[\mathcal{I}^*(s'), \mathcal{I}^*(t')]$, $\langle \mathcal{M}', \mathcal{I}' \rangle \models F'$,
2. $\mathcal{I}$' satisfies all the temporal constraints in $\mathcal{B}_i$

Different cases must be considered according to the applied expansion rule. The cases of the rules handling propositional connectives and temporal operators are similar to [7].

Let us consider the case of the δ-rule. Thus, in this case, our formula F has the form $\exists x A(x)$. By hypothesis, for any time point i in $[\mathcal{I}^*(s), \mathcal{I}^*(t)]$, $\mathcal{M}_i \models \exists x A(x)$, i.e., for any variable assignement σ, $\mathcal{M}_{\sigma,i} \models \exists x A(x)$. Thus, for any of such i, there exists an object $d_i \in D$ such that $\mathcal{M}_{\sigma',i} \models A(x)$, where σ' is the variant of σ giving the value d_i to the variable x. Let us say that d_i is the "witness" of $\exists x A(x)$ for the time point i. Let us consider the sequence of "witnesses" $\langle d_{j_0}, ...d_{j_n} \rangle$ respectively associated to the time points $j_0, ..., j_n$ of $[\mathcal{I}^*(s), \mathcal{I}^*(t)]$. If $d_{j_0} = d_{j_1} = ... = d_{j_n} = e$ for some e in the object domain D, then let $\mathcal{M}$' be constructed as follows. Say that $\mathcal{M} = \langle T, D, \delta_P, \delta_F \rangle$. Let δ'_F be such that $\delta'_F(f) = \delta_F(f)$ for any function symbol interpreted by $\mathcal{M}$ and $\delta'_F(a) = e$, where a is the new parameter appearing in the expansions of the δ-rule. Define $\mathcal{M}'$ as $\mathcal{M} = \langle T, D, \delta_P, \delta'_F \rangle$. Take $\mathcal{I}$'=$\mathcal{I}$. It is clear that, for any time point i in $[\mathcal{I}'^*(s), \mathcal{I}'^*(t)]$, $\mathcal{M}'_i \models A(a)$, so that the lemma is true for the leftmost expansion of the δ-rule. Otherwise, let j_p be the first time point such that $d_{j_p} \neq d_{j_{p+1}}$. Say that $d_{j_0} = d_{j_1} = ... = d_{j_p} = e$ for some $e \in D$. Define $\mathcal{M}$' exactly as in the previous case and extend $\mathcal{I}$ to $\mathcal{I}$' by setting $I'(c) = j_p$, where c is the new temporal constant appearing in the rightmost expansion. Clearly, for any time point i in $[\mathcal{I}'^*(s), \mathcal{I}'^*(c)]$, $\mathcal{M}'_i \models A(a)$, and for any time point j in $[\mathcal{I}'^*(c+1), \mathcal{I}'^*(t)]$, $\mathcal{M}'_j \models \exists x A(x)$. Moreover, $\mathcal{I}$' obviously satisfies $s \leq c$ and $c \leq t$. Thus, the rightmost expansion is satisfied.

For the γ rule, suppose that for any time point i in $[\mathcal{I}^*(s), \mathcal{I}^*(t)]$, $\mathcal{M}_i \models \forall x A(x)$, i.e., for any variable assignment σ, $\mathcal{M}_{\sigma,i} \models \forall x A(x)$. Thus, for any x-variant of σ (i.e. for any σ' such that $\sigma'(y) = \sigma(y)$ if y is not x), $\mathcal{M}_{\sigma',i} \models A(x)$. Let $A(t)$ the formula obtained expanding $\forall x A(x)$ and consider any expansion $\mathcal{M}$' of $\mathcal{M}$ to the parameters in t uninterpreted by $\mathcal{M}$ (if any). We have that $\mathcal{M}'_i \models A(t)$, as desired. The mapping $\mathcal{I}$' is $\mathcal{I}$ itself. Moreover, obviously $\mathcal{M}'_j \models \forall x A(x)$ for any $j \in [\mathcal{I}'^*(s), \mathcal{I}'^*(t)]$.

Finally, let us remark that the proof might be easily extended to the cases of the equality rules.

As a consequence, we have the following result.

Lemma 2. *Let $\mathcal{M} = \langle T, \sigma \rangle$ be a temporal interpretation, with $T = \langle 0, ..., n \rangle$, let S a finite set of closed LTL formulae in nnf and let K be the singleton $\{finish \leq start + k\}$, for some integer $k \geq n$. If $\mathcal{M}$ is a model of S, then any canonical tableau for $S \cup K$ is such that:*

1. *it has an open branch $\mathcal{B}$*
2. *there exists an extension $\mathcal{M}$' of $\mathcal{M}$ to the set of parameters P in $\mathcal{B}$ such that $\mathcal{B}$ is satisfiable in $\mathcal{M}$'.*

This immediately implies the folllowing theorem:

Theorem 1 (Soundness w.r.t. k-Unsatisfiability). *Let S be a set of closed formulae. If there exists a closed tableau for $S \cup \{finish \leq start + k\}$ then S is k-unsatisfiable. A fortiori, if there exists a finite closed tableau for $S \cup \{finish \leq start + k\}$, then S is k-unsatisfiable.*

Conversely, any complete open branch describes some model of its initial formulae. Such a result needs some preliminary lemmas and definitions. First of all, the invertibility of all the logical rules with respect to their logical nodes can be established.

The following lemma say that all the logical rules are invertible with respect to their logical nodes.

Lemma 3. *Let $\mathcal{M} = \langle T, \sigma \rangle$ be a temporal interpretation and $\mathcal{I}$ a temporal mapping with range T. For every logical rule*

$$\frac{\begin{array}{c}[s,t]\,F\\ s \leq t\end{array}}{\mathcal{B}_1 \qquad \mathcal{B}_2} \qquad or \qquad \frac{\begin{array}{c}[s,t]\,F\\ s \leq t\end{array}}{\mathcal{B}_1}$$

if $\langle \mathcal{M}, \mathcal{I} \rangle \models \mathcal{B}_i$ (for some $i = 1, 2$), then $\langle \mathcal{M}, \mathcal{I} \rangle \models [s,t]F$.

Proof. For the propositional and temporal rules, see []. The extension to the quantifier rules is straightforward.

The γ-rule enjoys also the following property:

Lemma 4. *Let $\mathcal{B}$ be an open and complete branch of a a tableau and let $[s, s']\forall x A(x)$ be a node in $\mathcal{B}$, expanded via a γ-rule. Let $\mathcal{M}$ be a temporal interpretation such that its object domain is exactly the set of ground terms in the branch, and $\mathcal{M}(t) = t$ for any ground term t in $\mathcal{B}$. Suppose that, for any labeled node $[s, s']A(t)$, where t is a ground term, for each i in $[I^*(s), \mathcal{I}^*(s')]$, $\mathcal{M}, i \models A(t)$. Then, for any such i, $\mathcal{M}, i \models \forall x A(x)$.*

Proof. Trivial.

The following obvious lemma holds:

Lemma 5. *Let $\mathcal{B}$ be a branch in a tableau and $\mathcal{I}$ a temporal mapping satisfying the constraints in $\mathcal{B}$. If $[s,t]A$ occurs in $\mathcal{B}$ and $\mathcal{I}^*(s) \leq \mathcal{I}^*(t)$, then $0 \leq I^*(s) \leq I^*(t) \leq \mathcal{I}(finish)$.*

The following definition captures the idea of a sequence of applications of the β-rule, each of which expands an expansion of the previous one, and the similar notion for the case of the δ-rule.

Definition 5. *Let $\mathcal{B}$ be a tableau branch. A β-node in $\mathcal{B}$ is a node of the form $[s,t]A \vee B$. A δ-node in $\mathcal{B}$ is a node of the form $[s,t]\exists x A(x)$. A β-chain in $\mathcal{B}$ is a sequence of β-nodes $X_0, X_1, ...$ such that, for every $i \geq 1$, X_{i-1} is expanded in $\mathcal{B}$ by application of the β-rule and X_i is the β-node in the corresponding rightmost expansion. A k-length-β-chain is a finite β-chain $X_0, X_1, ..., X_{k+1}$, constituted by $k+2$ nodes.*

A node X in $\mathcal{B}$ is the root node *of a β-chain if it is the first node in a β-chain and it is not itself obtained by an application of the β-rule, i.e. there exists a maximal length β-chain in $\mathcal{B}$ having X as its first node.*

The corresponding notions of δ-chain, k-lenght-δ-chain and root node of a δ-chain are defined similarly.

A tableau branch may contain an infinite number of β-chains or δ-chains. Moreover, in principle, each chain might be of unbounded length. For instance, if the branch contains a node $[s,t]\forall y \exists x A(y,x)$, an iterated application of the γ rule may produce a δ-chain whose root is $[s,t]\exists x A(t_1,x)$, a δ-chain whose root is $[s,t]\exists x A(t_2,x)$, etc, where the t_i are ground terms, and each of such chains might contain an unbounded number of nodes. However, if the branch is satisfiable by a model $\mathcal{M}$ whose temporal frame is $\langle 0,...,k\rangle$, each chain has at most $k+1$ elements. This fact is stated by the next lemma, whose proof is similar to the corresponding one in [7].

Lemma 6. *If $\mathcal{B}$ is a canonical open branch and its constraints have a solution $\mathcal{I}$ whose range is $\langle 0,...,n\rangle$, then $\mathcal{B}$ contains no k-length-β-chain or k-length-δ-chain with $k > n$.*

Proof. Let us assume that $\mathcal{B}$ does contain a k-length-β-chain with $k > n$. Then $\mathcal{B}$ contains a sequence of nodes having the form: $[s,t]\ A \vee B, [c_1+1,t]\ A \vee B, [c_2+1,t]\ A \vee B, ..., [c_{n+2}+1,t]\ A \vee B$ (where the c_i are temporal constants) and the sequence of constraints: $s \leq c_1, c_1+1 \leq c_2, ..., c_{n+1}+1 \leq c_{n+2}$, and $c_{n+2} \leq t$. If $\mathcal{I}$ is is a solution for the constraints in $\mathcal{B}$, then, by Lemma 5, $0 \leq \mathcal{I}^*(s) \leq \mathcal{I}^*(t) \leq \mathcal{I}^*(finish)$ (in fact $s \leq t$ is in $\mathcal{B}$), so that $\mathcal{I}^*(c_{n+2}) \geq n+1$. However, $\mathcal{I}^*(c_{n+2}) \leq \mathcal{I}^*(t) \leq \mathcal{I}^*(finish) \leq n$ holds too, that is absurd.

A similar reasoning can be done for the case of δ-chains.

The previous lemmas allow us to prove the following result:

Lemma 7. *If $\mathcal{B}$ is a complete and open tableau branch and $\mathcal{I}$ is a solution of the set of temporal constraints occurring in $\mathcal{B}$, then there is an interpretation $\mathcal{M}$ such that $\langle \mathcal{M}, \mathcal{I}\rangle \models \mathcal{B}$.*

Proof. Suppose that $\mathcal{I}$ is a solution for the constraints in $\mathcal{B}$ and let $T = \langle 0,...,n\rangle$ be the range of $\mathcal{I}$. Let $\mathcal{M}$ be $\langle T, D, \delta_P, \delta_F\rangle$ where:

- D is the set of ground terms in $\mathcal{B}$.

- for each i in T, sequence $t_1, .., t_n \in D^n$, and n-ary predicate p: $t_1, ..., t_n \in \delta_P(p)$ iff there is a node $[s,t]p(t_1, ..., t_n)$ in $\mathcal{B}$ such that $\mathcal{I}^*(s) \leq i \leq \mathcal{I}^*(t)$.
- For any n-ary function symbol f and i in T: $\delta_F(f)(t_1, ..., t_n) = f(t_1, ..., t_n)$.

We need to prove that, for any labelled formula $[s,t]F \in \mathcal{B}$ such that $[I^*(s), I^*(t)] \neq \emptyset$, and for any $i \in [I^*(s), I^*(t)]$: $\mathcal{M}_i \models F$ (the case of an empty interval being trivial). In order to do that, we define an order $<$ on the labeled formulae in $\mathcal{B}$ as follows. Given $[s,t]F \in B$ and $[s',t']F' \in \mathcal{B}$, we set $[s,t]F < [s',t']F'$ iff

- either F is a strict subformula of F'
- or else $[s,t]F$ is obtained by applying either a β-rule or a δ-rule or a $\mathcal{S}$-rule or an $\mathcal{U}$-rule to $[s',t']F'$.

Note that the order $<$ is partial and all the labeled formulae $[s,t]L$ in $\mathcal{B}$ where L is a literal are minimal for it. *The order $<$ is well founded*; in order to prove it, the only delicate point is the second item in the definition of $<$. In fact, a priori, there might be an infinite decreasing $<$-chain, whose starting point is either a β-formula or a δ-formula or a $\mathcal{S}$-formula or a $\mathcal{U}$-formula. However, since $\mathcal{I}$ is a solution to the constraints in $\mathcal{B}$ (with at most $n+1$ time points), Lemma 6 rules out such a possibility.

So, let us prove, by induction on $<$, that if $[s,t]F \in \mathcal{B}$ then, for any $i \in [\mathcal{I}^*(s), \mathcal{I}^*(t)]$, $\mathcal{M}_i \models F$.

- *Basis.*
  Here, F is a literal. If F is atomic, then the result holds by construction of $\mathcal{M}$. If F has the form $\neg p(t_1, .., t_n)$, suppose, by reductio, that there is some i in $[I^*(s), I^*(t)]$ such that $\mathcal{M}_i \not\models \neg p(t_1, .., t_n)$, i.e. $\mathcal{M}_i \models p(t_1, .., t_n)$. In this case, there is a node $[s',t']p(t_1, .., t_n)$ in $\mathcal{B}$ such that $i \in [\mathcal{I}^*(s'), \mathcal{I}^*(t')]$. An application of a conflict resolution rule would produce temporal constraints satisfiable by $\mathcal{I}$ only if $[\mathcal{I}^*(s), \mathcal{I}^*(t)] \cap [\mathcal{I}^*(s'), \mathcal{I}^*(t')] = \emptyset$, in contradiction with the fact that $i \in [I^*(s), I^*(t)$ and $i \in [\mathcal{I}^*(s'), \mathcal{I}^*(t')]$. Thus, $\mathcal{M}, i \models \neg p(t_1, .., t_n)$.
- *Inductive Step*
  We have 2 cases.
  1. The main operator is F is either a propositional connective or a temporal operator or else an existential quantifier. Since $\mathcal{B}$is complete, such a formula has been expanded in $\mathcal{B}$. The labeled formulae $[s',t']F'$ in the expansion are smaller than $[s,t]F$ according to the order $<$. Thus, by inductive hypothesis, for any i in $[\mathcal{I}^*(s')...\mathcal{I}^*(t')]$, $\mathcal{M}, i \models F'$. It suffices to apply the Lemma 3 to get the result.
  2. The main operator in F is $\forall$, i.e. F has the form $\forall x A(x)$. Since $\mathcal{B}$ is complete, such a formula has been expanded in $\mathcal{B}$ via the γ-rule, generating $[s,t]A(t)$ for each ground term t in $\mathcal{B}$. Each labeled formula $[s,t]A(t)$ is smaller than $[s,t]F$ according to $<$. Thus, by inductive hypothesis, for any i in $[\mathcal{I}^*(s)...\mathcal{I}^*(t)]$, $\mathcal{M}, i \models A(t)$. Then it suffices to apply Lemma 4 to get the result.

The proof may be slightly modified so as to hold also in the case where equality rules are present. Of course, in order to define the domain of the model $\mathcal{M}$, the set of ground terms needs to be quotiented w.r.t. equality. ■

As a consequence, we have the following result:

Theorem 2 (Quasi-Completeness w.r.t.k-unsatisfiability). *Let S be a set of formulae and k a natural number. If S is k-unsatisfiable, then there exists a closed tableau for $S \cup \{finish \leq start + k\}$*

The reason why we call such a result *quasi-completeness*, rather than completeness, is the following. As we have already remarked, a priori a closed branch migt be infinite, because, in principle, we could have an infinite set K of temporal constraints in the branch, such that any finite subset of K has a (finite) solution but K itself has no solution. Thus, in principle, we could have an infinite tableau for a formula A that is indeed closed without being able to finitely recognize this fact. Thus, if infinite closed tableaux might exists, Theorem 2 would not imply that the tableau calculus is an effective complete proof-system, allowing to semidecide the k-validity problem for $FO\text{-}LTL^{fin}$.

However, thanks to the presence of the initial constraint $finish \leq start + k$, *any closed branch is indeed finite.* In fact, if a branch contains an infinite set of constraints K and K has no solution, some finite subset of K has no solution. This property holds because:

1. Since the constraint $finish \leq start+k$ belongs to K, and, for each temporal constant c, the constraint $c \leq finish$ (implicitly) belongs to K, any solution for K is necessarily upper bounded by k.
2. If each finite subset of a denumerable set of constraints K has a solution that is upper bounded by some k, then K itselfhas a solution upper bounded by k.

The second fact is a corollary of a more general property:
Let K be a denumerable set of constraints and k a function from the set of constants C occurring in K to the natural numbers. If any finite subset K' of K has a solution $\mathcal{I}$ such that, for any constant $c \in K'$ $\mathcal{I}(c) \leq k(c)$, then K has a solution $\mathcal{I}^+$ such that, for each constant $c \in K$, $\mathcal{I}^+(c) \leq k(c)$.

Proof. This property can be proved by standard combinatorial techniques. Let $c_1, c_2, ...$ be an enumeration of the set C of constants appearing in K. Let $e_1, e_2, ...$ be an enumeration of the elements of K; such an enumeration induces an enumeration $K_1, K_2, ...$ of finite subsets of K, where $K_i = \{e_1, ..., e_i\}$. Say that the finite set of constants appearing in S_i is C_i. For each $i \geq 1$, by hypothesis there is some solution $\mathcal{I}_i$ of K_i such that $\mathcal{I}(c) \leq k(c)$ for any constant $c \in C_i$; let us extend such a solution $\mathcal{I}_i$ to a function $\mathcal{I}'_i$ from C to the natural numbers, by assigning any value to the constants in $C - C_i$. This operation can be visualized like the construction of a table, possibly having an infinite number of lines and an infinite numbers of columns:

	c_1	c_2	c_3	$\cdots$
K_1	$\mathcal{I}'_1(c_1)$	$\mathcal{I}'_1(c_2)$	$\mathcal{I}'_1(c_3)$	$\cdots$
K_2	$\mathcal{I}'_2(c_1)$	$\mathcal{I}'_2(c_2)$	$\mathcal{I}'_2(c_3)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

where the labels of the columns are the constants in C and the labels of the lines are the finite subsets K_i. Any line labeled by K_i corresponds to the codomain of the function $\mathcal{I}'_i$ and, for each j such that $c_j \in C_i$, the case (i, j) actually contains the value $\mathcal{I}_i(c_j)$, where $\mathcal{I}_i$ is a "local" solution for K_i. Given any j, let us call "OK cases" of the column j those cases (i, j) in the column such that (i, j) contains $\mathcal{I}_i(c_j)$. In order to obtain the required global solution $\mathcal{I}^+$, its values $\mathcal{I}^+(j)$ are defined by induction on j, as follows.
For $j = 1$, since the values appearing in all the OK cases of the first column are bounded by $k(c_1)$, and there are infinitely many such OK cases, there is at least a number n_1 which appears infinitely often in these cases. Set $\mathcal{I}^+(c_1) = n_1$, then update the table by erasing each line i such that the content of the case $(i, 1) \neq n_1$. (Note that an infinite numbers of lines is left).
For $j > 1$, the table updated at the previous stage is considered. Again, there is at least a number n_j which appears in all the infinitely many OK cases of the j-th column. Set $\mathcal{I}^+(c_j) = n_j$, then "erase" each line i such that the content of the case $(i, j) \neq n_j$
By construction, $\mathcal{I}^+$ satisfies each constraint in C, and, for any $c \in C$, $I^+(c) \leq k(c)$. ■

Hence, as a consequence of Theorems 1 and 2 we obtain:

Theorem 3. *Let S be a set of FO-LTL^{fin} formulae, k be any positive integer and $K = \{finish \leq start + k\}$.*

1. *If there exists a closed tableau for $S \cup K$, such a tableau is necessarily finite and S is k-unsatisfiable. (Soundness with respect to k-unsatisfiability)*
2. *If S is k-unsatisfiable, then there exists a finite closed tableau for $S \cup K$ (Completeness with respect to k-unsatisfiability)*

The tableau system defined in this work is not sound w.r.t. validity *tout court*: obviously, a formula A might be true in all the temporal interpretations whose time structure is bounded by a given number k without being valid. However, we can slightly modify the tableau definition, simply removing the initial constraint $finish \leq start + k$, to obtain a calculus that is indeed sound w.r.t. validity: if a closed tableau for $\neg A$ exists, then $\neg A$ is unsatisfiable, i.e. A is valid.

Theorem 4 (Soundness w.r.t. Unsatisfiability). *Let S be a set of formulae. If there exists a closed tableau for S then S is unsatisfiable. A fortiori, if there exists a finite closed tableau for S, then S is unsatisfiable*

Proof. Lemma 1 still holds for the unbounded calculus: its proof does not use the presence of the initial constraint. As a consequence, a result analogous to Theorem 2 holds: given any model $\mathcal{M}$ of S and any canonical tableau for S, there are an open branch $\mathcal{B}$ and an extension $\mathcal{M}$' of $\mathcal{M}$ to the set of parameters P in $\mathcal{B}$ such that $\mathcal{B}$ is satisfiable in $\mathcal{M}$. By contrapposition, if a closed tableau for S exists then S is unsatisfiable. ■

However, in the unbounded calculus, a closed tableau for S may contain an infinite branch. For instance, take S to be the unsatisfiable set of formulae:

$$\{\Box\exists x p(x)),\ \Diamond\forall y\neg p(y)\}$$

Because of the implicit contraction in the δ-rule, the rightmost branch of any tableau for S will contain an infinite set of constraints of the form $\{c_1+1 \leq c_2 \leq finish, c_2+1 \leq c_3 \leq finish, ...\}$: the absence of the initial constraint on the size of the searched model allows us to continue the construction of a δ-chain *ad libitum*. Each finite subset of such a set of constraints has a solution, but the set itself has no (finite) solution at all. Thus, although the tableau for S is closed, such a tableau is not a *refutation* of S.

Hence the unbounded calculus is *incomplete* w.r.t. to validity. Note that the analogous of Theorem 2 holds for the unbounded calculus: given any set S of unsatisfiable formulae, a closed tableau for S does exist. Yet, this does not provide a semi-decision procedure for validity.

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. N. Bidoit and S. DeAmo. A first step towards implementing dynamic algebraic dependencies. In *Lecture Notes in Computer Science*, volume 893, pages 308–319, Prague, Czech Republic, January 11-13 1995. Proceedings of 5th International Conference on Database Theory (ICDT'95), Springer Verlag.
6. S. Cerrito and M. Cialdea Mayer. First order linear temporal logic over finite time structures is not semi-decidable (draft). Available at `http://www.dia.uniroma3.it/~cialdea/papers/nonre.ps`.
7. S. Cerrito and M. Cialdea Mayer. Bounded model search in linear temporal logic and its application to planning. In H. De Swart, editor, *International Conference on Automated Reasoning with Analytical Tableaux and Related Methods*, pages 124–140. Springer, 1998.

8. S. Cerrito and M. Cialdea Mayer. Using linear temporal logic to model and solve planning problems. In F. Giunghiglia, editor, *Proceedings of the 8th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA'98)*. Springer Verlag, 1998.
9. R. Hähnle and O. Ibens. Improving temporal logic tableaux using integer constraints. In D. M. Gabbay and H. J. Ohlbach, editors, *Proceedings of the First International Conference on Temporal Logic (ICTL 94)*, volume 827 of *LNCS*, pages 535–539. Springer, 1994.
10. 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.
11. P. Wolper. The tableau method for temporal logic: an overview. *Logique et Analyse*, 28:119–152, 1985.