INFINITE TREES, MARKINGS AND WELL FOUNDEDNESS

by

R. Rinat, N. Francez, O. Grumberg

Technical Report #390

December 1985

TECHNION — ISRAEL INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

HAIFA, ISRAEL
INFINITE TREES, MARKINGS AND WELL FOUNDEDNESS

By

Ran Rinat, Nissim Francez and Orna Grumberg

Department of Computer Science
Technion- Israel Institute of Technology
Haifa, Israel.

December 1985

Abstract

A necessary and sufficient condition for a given marked tree to have no infinite paths satisfying a given formula is presented. The formulas are taken from a language introduced by Harel, covering a wide scale of properties of infinite paths, including most of the known notions of fairness. This condition underlies a proof rule for proving that a nondeterministic program has no infinite computations satisfying a given formula, interpreted over state sequences. We also show two different forms of seemingly more natural necessary and sufficient conditions to be inadequate.

1. Introduction

The problem of finding a sound and complete proof rule for proving that a given nondeterministic program terminates under a certain fairness assumption has been solved for various notions of fairness (e.g [AO 83, APS 84, FK 84, GFK 83, GFMR 81, LPS 81, PN 83]). In order for a program to be fairly terminating under any given notion of fairness, it has to admit no infinite fair computations, where the definition of a fair computation varies from one version of fairness to another and from one model of computation to another.

To be presented at the 6th CAAP conference, Nice, March 1986
A most convenient way of defining the semantics of a nondeterministic program is by using a tree, the vertices of which correspond to the intermediate states of computations. Thus, all these notions of fairness can be viewed as conditions on paths in the computation tree. Therefore, a language in which general conditions on paths in trees can be expressed can generalize all the already discussed types of fairness. One such a language, called $L$, is introduced in [HA 84]. It is also shown there how three notions of fairness can be expressed within $L$. Many of other versions of fairness, along with (what is claimed there as) any sensible condition one can think of, can also be expressed in $L$.

An important result of [HA 84] is a recursive transformation that, given a tree $T$ and a formula $\varphi \in L$, yields a tree $T'$, the infinite paths of which correspond to the infinite paths of $T$ satisfying $\varphi$. Thus, using this transformation, proving that $T$ has no infinite paths satisfying $\varphi$ reduces to proving that $T'$ has no infinite paths at all.

Returning to the issue of fair termination, two major approaches for proving it have been suggested in the literature (for a comprehensive discussion of these issues see [F 85]):

1. The method of helpful directions [GFMR 81, LPS 81]:

   According to this approach one defines a ranking of states by means of elements of a well-founded set. This ranking has to decrease according to rules derived directly from the fairness notion at hand.

2. The method of explicit scheduler [AO 83, APS 84, OA 84]:

   This approach is based on program transformation. By augmenting the given program using random assignments, an explicit fair scheduler is incorporated into the program. Thus, it remains to prove that the resulting program ordinarily terminates, for which a standard proof method exists.

   Thus, a natural goal is to provide generalizations of both methods to the context of languages like $L$: For any $\varphi \in L$, prove the absence (in a given program) of infinite computations satisfying $\varphi$. 
The application of the explicit scheduler method to $L$ is pursued in [DH 85], though they use a different kind of explicit scheduler than used in [AO 83]. In this paper we pursue the alternative approach of helpful directions, directly connecting the computation trees and their specifications in $L$ with decreasing well founded rankings. We deal here with a subset of $L$ called $L^-$, containing all formulas in $L$ having no infinite conjunctions or disjunctions. Weak and strong fairness can be expressed in $L^-$ ([HA 84]), but extreme fairness requires an infinite formula, and thus cannot be expressed in $L^-$. We remain in the level of trees, and the actual result concerns necessary and sufficient conditions, phrased in terms of decreasing well founded rankings, for a tree $T$ to have no infinite paths satisfying a formula $\varphi \in L^-$. This condition is intended to underly a syntactic proof rule for a programming language, having such trees as meaning for its programs.

Section 2 presents the language $L^-$. In section 3 the necessary and sufficient condition is presented. In section 4 we prove its correctness, and in section 5 we show the impossibility of two, seemingly simpler and more natural, forms of necessary and sufficient conditions.

2. Basic definitions

We first define the trees, to which we refer. A node is a finite sequence of natural numbers (i.e. an element of $N^*$) and a tree is a set of nodes (i.e. a subset of $N^*$) closed under the prefix operation. The root of the tree is $\varepsilon$, and a path is a maximal increasing sequence of successive nodes (by the prefix ordering) starting at $\varepsilon$. Parts of a path are termed path-\textit{fragments}, or just \textit{fragments}. A node is a leaf if it is the last element of a (finite) path. An example of a tree is shown in figure 1. A tree is well founded if all its paths are finite.

Let $\Sigma$ be some fixed (possibly infinite) alphabet. A $\Sigma$-\textit{marked tree} is one in which nodes are marked with (possibly infinitely many) letters from $\Sigma$, i.e. a tree $T$ comes complete with a marking predicate $M_T \subseteq T \times \Sigma$.

Throughout this paper, we refer to recursive marked trees, i.e marked trees for which two algorithms exist: one that given an element of $N^*$ decides whether it is a
Figure 1. An example of a tree.

node in the tree and of which kind (leaf or internal). The other decides, given a node \( v \) in the tree and a mark \( a \), whether \( v \) is marked with \( a \). Keeping in mind that the trees are to be regarded as denotations of programs, we discuss only recursive trees, and thus we do not specify this explicitly unless needed.

We now define the language \( L^- \) for stating properties of infinite paths in a marked tree (we repeat the definition of \( L \) from [HA 84] omitting infinite disjunctions and conjunctions). An atomic formula is an expression of one of the forms \( \exists_a \), \( \forall_a \), \( \exists \) \( a \) or \( \forall \) \( a \), where \( a \in \Sigma \) is a mark. \( L^- \) is the closure of the atomic formulas under finite conjunctions and disjunctions. Note the absence of negation from \( L^- \) (and \( L \)).

We interpret the formulas of \( L^- \) over infinite paths in a marked tree as follows: Let \( \varphi \) be an atomic formula and let \( \pi \) be an infinite path. \( \pi \) satisfies \( \varphi \) (\( \pi \models \varphi \)) if either

(a) \( \varphi = \exists a \) and there is a node on \( \pi \) marked with \( a \).

(b) \( \varphi = \forall a \) and all the nodes on \( \pi \) are marked with \( a \).
(c) $\varphi = \exists a$ and there are infinitely many nodes on $\pi$ marked with $a$.

(d) $\varphi = \forall a$ and there is a vertex in $\pi$ from which all the nodes are marked with $a$.

The interpretation of a general formula is obtained from the atomic formulas using the usual meaning of the boolean connectives. Note that formulas are interpreted over infinite paths only, and so the notion of a finite path satisfying a formula is undefined.

For example, consider playing chess on an infinite board (but with the standard set of 32 pieces) where moving rules are generalized in some reasonable way. An infinitely long game is a draw if both players call "check" infinitely often, otherwise it is a win for the player with the most calls. The game tree can be regarded as a marked tree with, say $a$ and $b$ marking nodes where player 1 or 2 checks respectively. The draw criterion is then given simply by the formula of $L^- : \exists a \land \exists b$.

We say that a tree $T$ is $\varphi$-avoiding, or that $\text{avoid}(T,\varphi)$ holds, if $T$ has no infinite paths satisfying $\varphi$.

3. The proof rule

Our goal is to present a necessary and sufficient condition for a given marked tree to have no paths satisfying a given formula, based on a decreasing well founded ranking. Building on such condition, a higher level goal is to present a (sound and semantically complete) proof rule for proving that a nondeterministic program (having a tree as its operational semantics) has no infinite executions satisfying a given formula (properly interpreted). Keeping this higher goal in mind, we metaphorically refer to the condition itself as a proof rule. We would like the rule to follow the following scheme: given a marked tree $T$ and a formula $\varphi \in L^-$, for which we want to prove $\text{avoid}(T,\varphi)$, choose a well founded set $(W,\prec)$, and a variant (or rank) function $\rho$, mapping nodes in $T$ to $W$. The variant $\rho$ should satisfy certain restrictions, assuring that the tree $T$ is $\varphi$-avoiding, i.e such $W$ and $\rho$ should exist if the tree is $\varphi$-avoiding.

These restrictions are, generally speaking, as follows: First, we require that the rank be nonincreasing, i.e if a node is mapped to some value then all of its descendents are mapped to lower or equal values. Second, we require that whenever a node's marking "leads" to the satisfaction of $\varphi$, in a sense to be made precise, all of its descendents are
mapped to strictly lower values. The determination whether a node leads or does not lead to the satisfaction of $\varphi$ is formalized by means of an appropriate predicate on the nodes, providing this information. This predicate, which depends on the given formula, has to be such, that an infinite path contains infinitely many nodes satisfying the predicate iff this path satisfies the formula. Also, keeping in mind that this rule has to be applicable to programs, we would like the predicate to depend only on the markings of the nodes on the fragment from the root to the node itself.

However, the rule we present is of a more complicated form. The reason for this complication is that a rule of the above form is impossible, since no appropriate predicate can be found. This fact is made precise and proved in a later section. Another, more complicated, form of a rule, still more natural than the one we are about to present, is also shown to be impossible.

The difference between the simple scheme discussed above and the rule to be actually presented is, that instead of finding one well founded ranking for the whole tree, we have to find one for each node in $T$. Then, we use the same technique, but instead of referring to the whole tree, we refer to each subtree of $T$. Given a vertex $v$ in the tree and a node $u$ in the subtree of $T$ rooted at $v$, we have the corresponding node-predicate, telling whether $u$ leads to the satisfaction of $\varphi$ in the subtree rooted at $v$. The exact meaning of this is explained after the definition of the predicate. Due to the nature of the rule, this predicate has to depend not only on the fragment from $v$ to $u$, but also on the fragment from the root (of $T$) to $v$. There is also a need to distinguish between these two parts of the fragment from the root to $u$, and therefore the predicate has two arguments. Since its values depend only on markings found on a path, its arguments are elements of $(2^E)^*$, where each symbol $x \in 2^E$ represents a set of marks from $\Sigma$.

**Definitions:** Let $\Sigma$ be an alphabet of marks and let $\pi \in (2^E)^*$.

1. A mark $a$ appears in $\pi$ if there exist $x \in 2^E$ and $u,v \in (2^E)^*$, such that $\pi=uxv$ and $a \in x$. If $v=\varepsilon$ then $a$ appears in the last element of $\pi$. 
2. A mark $a$ appears in all $\pi$ if $\pi = x_1 \ldots x_n$, $x_i \in 2^\Sigma$ and $a \in x_i$ for all $1 \leq i \leq n$.

We now proceed to the definition of the predicate $Q$, which, intuitively, given the markings on the fragment from the root to $v$, and the markings on the fragment from $u$ to $v$, tells whether $u$ leads to the satisfaction of $\varphi$ in the subtree rooted at $v$.

Let $\varphi \in L^*$ be a formula over $\Sigma$. Define the predicate $Q_{\varphi}(\Sigma; (2^\Sigma)^* \times (2^\Sigma)^* \cup \{T, F\})$ inductively as follows (we omit $\Sigma$ and write $Q$ instead of $Q_{\varphi}$ where $\Sigma$ is understood from the context):

**Basis:**

- $Q_{\exists a}(\pi_1, \pi_2)$ iff $a$ appears in $\pi_1 \pi_2$.
- $Q_{\forall a}(\pi_1, \pi_2)$ iff $a$ appears in all $\pi_1 \pi_2$.
- $Q_{\exists a}(\pi_1, \pi_2)$ iff $a$ appears in the last element of $\pi_1 \pi_2$.
- $Q_{\forall a}(\pi_1, \pi_2)$ iff $a$ appears in all $\pi_2$.

**Inductive step:** Assuming that for $\varphi, \psi \in L^*$, $Q_\varphi(\pi_1, \pi_2)$ and $Q_\psi(\pi_1, \pi_2)$ have been defined for every $\pi_1$ and $\pi_2$, we have:

- $Q_{\varphi \land \psi}(\pi_1, \pi_2)$ iff $Q_\varphi(\pi_1, \pi_2)$ or $Q_\psi(\pi_1, \pi_2)$.

The definition of $Q_{\varphi \land \psi}(\pi_1, \pi_2)$ involves induction on the length of $\pi_2$:

- $Q_{\varphi \land \psi}(\pi_1, \varepsilon)$ holds for all $\pi_1$.

Let $\pi_2 = \bar{\pi}_2 x$ where $x \in 2^\Sigma$. Assume $Q_{\varphi \land \psi}(\pi_1, \pi_2')$ has been defined for every $\pi_2'$ which is a prefix (not necessarily a proper one) of $\pi_2$. Let $\bar{\pi}_2$ be the longest prefix of $\pi_2'$ such that $Q_\varphi(\bar{\pi}_2, \pi_2)$ holds (such a prefix always exists because of the definition for $\pi_2 = \varepsilon$). Then we have:

- $Q_{\varphi \land \psi}(\pi_1, \pi_2)$ iff there exist $\pi$, $\pi', \varepsilon$ such that $\bar{\pi}_2 \pi$ and $\bar{\pi}_2 \pi'$ are prefixes of $\pi_2$ and both $Q_\varphi(\pi_1, \bar{\pi}_2 \pi)$ and $Q_\varphi(\pi_1, \bar{\pi}_2 \pi')$ hold.

We now explain the intuition behind this definition. The argument $\pi_1$ of $Q$ is to be identified with the word along the fragment from the root of $T$ to $v$, which is the root of
the subtree in mind. Similarly, $\pi_2$ is to be identified with the fragment from $v$ to the specific node $u$, for which $Q$ has to determine whether it leads to the satisfaction of $\varphi$.

Now, for all atomic formulas but $\forall_a$, $Q$ refers to the concatenation of its two arguments. Hence for a given $u$, the determination whether it leads to satisfaction does not depend on $v$, i.e., it is the same for all subtrees containing $u$.

For all the first three atomic formulas, $Q$ determines leading for satisfaction in a natural way. For $\forall_a$, $Q$ ignores the first argument, and relates only to the fragment from $v$ to $u$, thus relating only to the subtree rooted at $v$. Restricted to this subtree, $Q$ behaves as if the formula at hand is $\forall_a$. This fact can be justified by noticing, that a tree satisfies $\forall_a$ iff all its subtrees satisfy $\forall_a$. Thus, "leading to satisfaction within a given subtree" gets its meaning only through this formula, and the meaning is, actually, leading to the satisfaction of $\forall_a$ in this subtree. Of course, all this is extended when connectives are introduced.

The definition of $Q$ for disjunction is straightforward. In the conjunction case, $Q$ always holds for the root of the subtree. For all other nodes $u$, $Q$ holds in $u$ iff there are nodes on the fragment from $u$'s most recent ancestor for which $Q$ holds, for which $Q$ has determined leading to satisfaction of the two conjuncts of $\varphi$. Thus, for a given path, $Q$ will induce infinitely many decrements iff there are infinitely many pairs of nodes for which $Q$ has induced decrement with respect to the conjuncts of the conjunction. This happens iff $Q$ has induced infinitely many decrements for each conjunct.

It is important to note that $Q$ is recursive, i.e., there is an algorithm to calculate $Q$ from its two arguments.

Notations: Let $T$ be a $\Sigma$-marked tree.

1. If $v, u$ are vertices in the tree connected by $v = v_1 - \ldots - v_{n-1} - u$, denote by $\pi_{v,u}$ the following element of $(2^\Sigma)^*$: $\pi_{v,u} = z_1 \cdots z_n$, where $z_i$ is the set of marks of $u_i$. If $v$ is the root of the tree, we write $\pi_u$ instead of $\pi_{v,u}$.

2. For a vertex $v \in T$, denote by $T_v$ the subtree of $T$ rooted in $v$.

3. For a vertex $v$, denote by $\text{children}(v)$ the set of all (direct) children of $v$. 
4. For a vertex $v$, the predicate $\text{leaf}(v)$ holds iff $v$ is a leaf.

We can now present the rule:

Let $T$ be a $\Sigma$-marked tree, and let $\phi \in L^-$ be a formula over $\Sigma$. To prove $\text{avoid}(T, \phi)$, choose for each $v \in T$ a well founded set $(W_v, \prec_v)$ and a predicate $P_v: W_v \times T_v \rightarrow \{T, F\}$ such that the following conditions hold:

1. There exists $w \in W_v$ such that $P_v(w, v)$ holds.
2. For all $u \in T_v$ and for all $w \in W_v$,
   
   (a) $P_v(w, u) \land Q_v(\pi_v, \pi_{v, u}) \rightarrow \forall s \in \text{children}(u) \exists w' < w: P_v(w', s)$.
   
   (b) $P_v(w, u) \rightarrow \forall s \in \text{children}(u) \exists w' \leq w: P_v(w', s)$.
3. For all $u \in T_v$, $\text{leaf}(u) \iff P_v(0_v, u)$.

A proof rule for proving that $T$ is $\phi$-avoiding.

Due to technical reasons, we have chosen to use the parametrized predicate $P$ instead of a variant function assigning values to nodes. Clause (1) of the rule assures that each root of a subtree has some value assigned to it. Clause (2)(a) assures that whenever $Q$ induces decrement in a node $u$ of a subtree rooted at $v$, then all of $u$'s children are assigned strictly lower values. Clause (2)(b) assures that the values are nonincreasing. Clause (3) is included due to technical reasons, which will be effective when the rule is applied to programs. Its meaning is that a node is assigned $0_v$ iff it is a leaf. Here $0_v$ is a generic name for the minimal element of $W_v$.

Note that since $T$ and $Q$ are recursive, then there is an algorithm which decides, given a node in the tree, whether its children should decrease in rank.

Example: Consider the tree $T$ shown in figure 2(a), where all nodes but the ones on the
"main" path (the infinite one) are marked with $a$, and the formula $\varphi = \exists w x_a$. Clearly, $T$ is $\varphi$-avoiding since its only infinite path is not marked at all. To prove it, choose $W_v$ to be the ordinal $\omega + 1$ with the usual $\in$-ordering for all $v \in T$. For a node $u$ in $T_v$, let $P_u(\omega, u) = T$ if $u$ is on the main path, and $P_u(\eta_u, u) = T$ if $u$ is not on the main path, and $\eta_u$ is the distance between $u$ to a leaf. The ranking for the root of $T$ is shown in figure 2(b). It is easy to see that conditions (1)-(3) hold with the chosen $P_v$. Note that this example also demonstrates the fact, that the natural numbers ($\omega$) do not always suffice, even when the tree is of a finite (and even bounded) degree.

4. Soundness and completeness of the rule

Throughout this section we let $T$ be some fixed $\Sigma$-marked tree.

**Definition:** Let $\pi$ be an infinite path in $T$, $v$ a vertex on $\pi$, and $\varphi \in L^-$ a formula.

![Diagram](image.png)

**Figure 2.** An example of a ranking function.
\[ \pi \text{ is } \varphi\text{-decreasing starting at } v \text{ if there is an infinite sequence of nodes on } \pi, \]
\[ u_1, u_2, u_3, \ldots \text{ s.t. } u_1 \text{ is a descendent of } v, \text{ each } u_i \text{ is a descendent of } u_{i-1}, \text{ and for all } i, \]
\[ Q(\pi_v, \pi_{uv_i}) \text{ holds. We say that } \Delta(\pi, \varphi, v) \text{ holds to express the above notion. The sequence } u_1, u_2, u_3, \ldots \text{ is called a } \text{decreasing sequence.} \]

**Lemma 1:** Let \( \pi \) be an infinite path in \( T \), \( \varphi \) a formula, and \( v, v' \) vertices on \( \pi \) s.t. \( v' \) is a descendent of \( v \). If \( \Delta(\pi, \varphi, v) \) holds then so does \( \Delta(\pi, \varphi, v') \).

**Proof:** By induction on the structure of \( \varphi \).

(a) \( \varphi = \exists a \). Let \( u_1, u_2, \ldots \) be a decreasing sequence causing \( \Delta(\pi, \exists a, v) \) to hold. By the definition of \( Q \), this means that each \( u_i \) is marked \( a \). Let \( j \) be the minimal number s.t. \( u_j \) is a descendent of \( v' \). Then \( Q(\exists a, \pi_v, \pi_{uv_k}) \) holds for all \( k \geq j \), and so \( \Delta(\pi, \varphi, v') \) holds.

(b) \( \varphi = \exists a \). Let \( u_1, u_2, u_3, \ldots \) be as in (a). Since \( Q(\exists a, \pi_v, \pi_{uv_k}) \) holds for all \( i \), then \( a \) appears in \( \pi_v \pi_{uv_k} \) for all \( i \). Let \( j \) be as in (a). Clearly, \( a \) appears in \( \pi_v \pi_{uv_k} \) for all \( k \geq j \), and thus \( Q(\exists a, \pi_v, \pi_{uv_k}) \) holds, and consequently so does \( \Delta(\pi, \varphi, v') \).

(c) \( \varphi = \forall a \). The proof is similar to (a) and (b).

(d) \( \varphi = \forall a \). The proof is again similar to (a) and (b) (the fact that \( Q(\forall a, \pi_{1}, \pi_{2}) \) depends on \( \pi_2 \) only is immaterial).

**Induction step:**

(a) \( \varphi = A \lor B \). Let \( u_1, u_2, u_3, \ldots \) be as in the basis proofs. By the definition of \( Q \) and w.l.o.g we can assume that there is a sequence \( i_1, i_2, \ldots \) s.t. \( Q(\pi_v, \pi_{u_{i_j}}) \) holds for all \( j \), and so \( \Delta(\pi, A, v) \) holds. By the induction hypothesis, so does \( \Delta(\pi, A, v') \). By the definition of \( Q \), \( \Delta(\pi, \varphi, v') \) holds.

(b) \( \varphi = A \land B \). Let \( u_1, u_2, u_3, \ldots \) be as in (a). By the definition of \( Q \), each \( u_i \) is associated with a pair of nodes \( s, s' \) s.t. \( Q(\pi_v, \pi_{uv_s}) \) and \( Q(\pi_v, \pi_{uv_{s'}}) \) hold (actually, either \( s' \) or \( s' \) must be \( u_i \)). Therefore, both \( \Delta(\pi, A, v) \) and \( \Delta(\pi, B, v) \) hold. By the hypothesis, the same is true for \( v' \). Thus, there are infinitely many pairs of nodes
Lemma 2: Let \( \pi \) be an infinite path in \( T \) and let \( \varphi \) be a formula. \( \pi \) satisfies \( \varphi \) iff there is a vertex \( v \) on \( \pi \) s.t. \( \Delta(\pi, \varphi, v) \) holds.

Proof: Assume that \( \pi \) satisfies \( \varphi \). The proof is by induction on the structure of \( \varphi \).

Basis:

(a) \( \varphi = \exists_a \). Choose \( v \) to be the root (the first vertex of \( \pi \)).

(b) \( \varphi = \exists_a \). Choose \( v \) to be the root.

(c) \( \varphi = \forall_a \). Choose \( v \) to be the root.

(d) \( \varphi = \forall_a \). Choose \( v \) to be the first vertex on \( \pi \) from which all vertices are marked \( a \).

Induction step:

(a) \( \varphi = A \land B \). Assume w.l.o.g that \( \pi \) satisfies \( A \). By the hypothesis, let \( v \) be a vertex s.t. \( \Delta(\pi, A, v) \) holds. By the definition of \( Q \), so does \( \Delta(\pi, A \land B, v) \).

(b) \( \varphi = A \lor B \). Let \( v_A, v_B \) be vertices in \( \pi \) s.t. \( \Delta(\pi, A, v_A) \) and \( \Delta(\pi, B, v_B) \) hold (by the hypothesis). Assume w.l.o.g that \( v_B \) is a descendent of \( v_A \). By lemma 1, \( \Delta(\pi, A, v_B) \) holds, and so, by \( Q \)'s definition, so does \( \Delta(\pi, A \lor B, v_B) \). Thus, choose \( u \) to be \( v_B \).

For the other direction, assume that \( \Delta(\pi, \varphi, v) \) holds for some \( v \) in \( \pi \). The proof is again by structural induction on \( \varphi \).

Basis:

(a) \( \varphi = \exists_a \). By the assumption, there is a sequence \( u_1, u_2, u_3 \ldots \) s.t. \( Q_{\exists_a}(v, v, u_i) \) holds for all \( i \). This implies that each \( u_i \) is marked \( a \), and thus \( \pi \) satisfies \( \varphi \).

(b) \( \varphi = \exists_a \). Let \( u_1, u_2, u_3 \ldots \) be as in (a). By definition, \( a \) appears in \( v, v, u_i \), and so \( \pi \) satisfies \( \varphi \).

(c) \( \varphi = \forall_a \). Let \( u_1, u_2, u_3 \ldots \) be as in (a). We have to show that each node in \( \pi \) is marked \( a \). Let \( u \) be such a node. Let \( u_j \) be some vertex from the decreasing sequence appearing after \( u \) (i.e. \( u_j \) is a descendent of \( u \)). Since \( Q_{\forall_a}(v, v, u_j) \)
holds, we have that \( a \) appears in all \( \pi_v \pi_w \pi_j = \pi_w \). Since \( u \) is a node on \( \pi_j \), we have that \( u \) is marked \( a \).

d) \( \varphi = \forall a \). By an argument similar to the previous one, we have that starting at \( v \), \( \pi \) is everywhere marked \( a \), and therefore \( \pi \) satisfies \( \varphi \).

Induction step:

(a) \( \varphi = A \lor B \). By the assumption, by \( Q \)'s definition and w.l.o.g we can assume that \( \Delta(\pi, A, v) \) holds. By the induction hypothesis, \( \pi \) satisfies \( A \) and thus also satisfies \( A \lor B \).

(b) \( \varphi = A \land B \). By \( Q \)'s definition both \( \Delta(\pi, A, v) \) and \( \Delta(\pi, B, v) \) hold. By the hypothesis \( \pi \) satisfies \( A \) and satisfies \( B \), thus also \( A \land B \).

Theorem 1 (soundness of the proof rule): Let \( \varphi \in L^- \) be a formula. If for all \( v \in T \) there is a well founded set \((\mathcal{W}_v, <_v)\) and a predicate \( P_v : \mathcal{W}_v \times T_v \to \{T, F\} \) s.t (1),(2) and (3) are satisfied then \( \text{avoid}(T, \varphi) \) holds.

Proof: Assume that such \( \mathcal{W}_v \) and \( P_v \) exist for each \( v \in T \), and Assume , by way of contradiction, that there is an infinite path \( \pi \) in \( T \) that satisfies \( \varphi \). By lemma 2, there is a vertex \( v \) on \( \pi \) s.t \( \Delta(\pi, \varphi, v) \) holds. Let \( u_1, u_2, u_3 \ldots \) be a decreasing sequence for which the above is true. By (1), there exists \( w \in \mathcal{W}_v \) s.t \( P_v(\pi, w) \) holds. By (2) and (3) we have that if \( P_v(\pi, w) \) holds for some \( w \) and \( i \), then there exists \( w' < w \) s.t \( P_v(w', u_{i+1}) \) holds, and this fact is still valid taking \( v \) as \( u_0 \). Thus, for each \( u_i, u_j \) s.t \( i > j \) there are \( u_i, u_j \) s.t \( u_i < u_j \) and \( P_v(u_i, u_j) \). The sequence \( u_1, u_2 \ldots \) is, therefore, an infinite decreasing sequence of elements from \( \mathcal{W}_v \) - a contradiction to the well foundness of \( \mathcal{W}_v \).

Theorem 2 (completeness of the proof rule): Let \( \varphi \in L^- \) be a formula. If \( \text{avoid}(T, \varphi) \) holds then For each \( v \in T \) there exist \((\mathcal{W}_v, <_v)\) and \( P_v : \mathcal{W}_v \times T_v \to \{T, F\} \) satisfying (1),(2) and (3).
Proof: Assume that $avoid(T, \varphi)$ holds. We have to find appropriate $W_v$ and $P_v$ for each $v \in T$.

Let $v \in T$ be a vertex. Call a vertex $u$ in $T_v$ decreasing if $Q_\varphi(\pi_v, \pi_{\nu,v}, u)$ holds and steady otherwise. From lemma 2 it follows that in $T_v$ there is no infinite path with infinitely many decreasing nodes (since this would imply that this path satisfies $\varphi$, and thus $avoid(T, \varphi)$ would not hold).

Definition:

1. Let $u$ be a vertex in $T_v$. $cone(u)$ is the subtree of $T_v$ rooted in $u$, truncated after each decreasing node, i.e $cone(u)$ includes all path-fragments starting at $u$ up to, and including, a decreasing node. If a fragment, finite or infinite, does not contain any decreasing nodes, then it is wholly contained in the cone. Note that $u$ itself is always contained in $cone(u)$, thus a cone is never empty.

2. An exit node from a cone is a child of a decreasing node, after which the cone is truncated.

The above definition is illustrated in figure 3. Decreasing nodes are darkened.

such $u_1, u_2, u_3, u_4$ and $u_5$ are exit nodes from $cone(v)$.

![Figure 3. Cones and exit nodes.](image-url)
Now, inductively construct from $T_v$ the tree $T_v^*$, having cones as nodes as follows:

- The root of $T_v^*$ is $cone(u)$.
- Assume that $cone(u)$ is a node in $T_v^*$ for some $u \in T_v$.
  - If $cone(u)$ has no exit nodes then it is a leaf in $T_v^*$.
  - If $cone(u)$ has exit nodes $u_1, u_2, u_3 \ldots$ (possibly countably many) then each $cone(u_i)$ is a child of $cone(u)$ in $T_v^*$.

Observations:

1. If $cone(u')$ is a child of $cone(u)$ then there is a path-fragment in $T_v$ from a decreasing node in $cone(u)$ to $u'$.

2. From 1 it follows that $T_v^*$ is a tree (i.e., it is acyclic) since a cycle in $T_v^*$ would imply one in $T_v$.

3. From 1 it also follows, that there is no infinite path in $T_v^*$, since this would imply an infinite sequence of decreasing nodes in $T_v$.

4. Each node in $T_v$ is included in exactly one cone in $T_v^*$.

5. Just as in section 2.

Now, since $T_v^*$ is well founded, there is a well-founded set $W_v$ that can rank $T_v^*$ in such a way that a parent has always a greater rank than each of its children. Also, without loss of generality, we can assume that no vertex in $T_v^*$ is ranked $0_v$ (the minimal element of $W_v$).

Denote the ranking of a vertex $u$ in $T_v^*$ by $rank(u)$. We can now find $P_v: W_v \times T_v \rightarrow \{T, F\}$ could define as required. For all $u \in T_v$ and $w \in W_v$ we define $P_v$ ($w, u$) = $T$ $\iff$ (leaf ($u$) and $w = 0_v$) or ($rank(c[u]) = w$).

Alternatively, where $c[u]$ is the cone in $T_v^*$ to which $u$ belongs.

We now have to verify that (1), (2) and (3) hold with $W_v$ and $P_v$:

1. There are
2. For all

(a) $P_v(w, u) \land Q_v(\pi^v, \pi^v, u)$ means that $u$ is decreasing and either $rank(c[u]) = w$, or $u$ is a leaf. If $u$ is a leaf then the condition holds vacuously. Otherwise, each child $u'$ of $u$ is an exit node from $c[u]$, and therefore belongs to another cone, $cone(u')$, which is a child of $c[u]$. By the main characteristic of the ranking
function, it follows that \( \text{rank}(\text{cone}(u')) < \text{rank}(c[u]) = w \), and thus \( w' = \text{rank}(\text{cone}(u')) \) satisfies the requirements in case \( u' \) is not a leaf. If \( u' \) is a leaf, then \( w' = 0 \), will do.

(b) \( P_v(w, u') \) means that \( \text{rank}(c[u]) = w \), or \( u' \) is a leaf. The leaf case leads again to vacuous satisfaction. Otherwise, if \( u \) is decreasing then the case is as in (a). If not, then each child of \( u \) is still included in \( c[u] \), and therefore is either a leaf and is ranked \( 0_v \), or has the same rank as \( u \). At any case, the requirements are satisfied.

5. If \( u \) is not a leaf, then by the assumption on \( W_0 \) (that no node in \( T_0 \) is ranked \( 0 \)), we have that \( P_v(0_v, u) \) does not hold.

Lemma 5.1

If \( u \) is a leaf then by definition \( P_v(0_v, u) \) holds.

Proof of Lemma 5.1

5. Justification of the proof rule

As mentioned before, the suggested rule might seem more complicated than necessary, since it requires finding a well founded set for each \( v \in T \). Moreover, we are led to define \( Q \) having two arguments instead of one. It would be more natural if we could define a predicate \( R_{\epsilon, \phi}(\mathcal{E}) \rightarrow \{T, F\} \) in some way, adhering to the following proof rule:

Alternative proof rule 1: To prove \( \text{avoid}(T, \phi) \), find a well founded set \( (W, <) \) and a predicate \( P : W \times T \rightarrow \{T, F\} \) s.t the following conditions hold:

\[ (1) \text{ There exists } w \in W \text{ such that } P(w, \text{root}) \text{ holds.} \]

\[ (2) \text{ For all } v \in T \text{ and for all } w \in W, \]

\[ (a) P(w, u) \land R_v(p_v) \rightarrow \forall s \in \text{children}(v) \exists w' < w : P(w', s). \]

Proof:

\[ (b) P(w, u) \rightarrow \forall s \in \text{children}(v) \exists w' \leq w : P(w', s). \]
(condition (3) (the leaf condition) is omitted here since it was introduced for technical reasons only).

Unfortunately, however natural this rule is, it is impossible, since an appropriate predicate $R$ cannot be found. We now prove that.

**Theorem 3:** There exists no recursive predicate $R$, with which the alternative proof rule is sound and complete.

**Proof:** By way of contradiction, let $R$ be such a predicate. In order to proceed we need the following lemma:

**Lemma 3:** Let $T$ be a tree composed of one path only, $v_1, v_2, v_3, \ldots$ where $v_i$ is the descendant of $v_{i-1}$, and let $\varphi$ be a formula. $T$ satisfies $\varphi$ if and only if there are infinitely many vertices $v$ on $T$ for which $R_\varphi(v)$ holds (we refer to such nodes as decreasing with respect to $\varphi$).

**Proof (of lemma 3):** Assume that $T$ satisfies $\varphi$, and assume, by way of contradiction, that there are only finitely many decreasing nodes on $T$. Let $k$ be the maximal index $s.t. u_k$ is decreasing. Choose the well founded set $W = \{0, 1, \ldots k\}$ and the predicate $P: W \times T \rightarrow \{T, F\}$ defined as follows:

$$P(w, u_i) = T \iff (w = k + 1 - i \text{ and } 1 \leq i \leq k) \text{ or } (w = 0 \text{ and } i > k).$$

It is easy to see that conditions (1) and (2) hold with $W$ and $P$, which is a contradiction to the soundness of the rule.

Now, assume that there are infinitely many decreasing nodes in $T$, $v_1, v_2, \ldots$. We show that there exist no $W$ and $P$ which satisfy the conditions of the proof rule. By contradiction, assume their existence. From conditions (1) and (2) it follows, that for all $j \geq 1$ there exists $w_j$ s.t $P(w_j, v_j)$ holds, and $w_j > w_{j+1}$. Therefore, the sequence $w_1, w_2, \ldots$ is an infinite decreasing sequence of elements from $W$ - a contradiction.

From the completeness assumption we have that $\text{avoid}(T, \varphi)$ does not hold, hence $T$ satisfies $\varphi$. 

\[ \text{\square} \]
We now proceed with the proof of theorem 3. Let $T_0$ be the following tree:

$$T_0: 0--0--0--0--0--0$$

i.e. a path everywhere marked by $a$. $T_0$ satisfies $\forall a$, and so, by lemma 3, there are infinitely many decreasing nodes on $T$ with respect to $\forall a$ (i.e. there are infinitely many nodes $v$, for which $R_{\forall a}(\pi_v)$ holds). Let $v_1$ be the first such node (the closest to the root), and let $u_1$ be its successor. The tree $T_1$ is the following:

$$T_1: 0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0$$

i.e. $T_1$ has the same vertices as $T_0$, only they are marked differently, and the difference is that $u_1$ has no mark. From the above, we have that $v_1$ is decreasing with respect to $\forall a$. Also, $T_1$ satisfies $\forall a$ and thus, by lemma 3, there are infinitely many decreasing nodes in it. Let $v_2$ be the first such node after $v_1$, and let $u_2$ be its successor. The tree $T_2$ is the following:

$$T_2: 0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0$$

Again, we have that $v_1$ and $v_2$ are decreasing with respect to $\forall a$, and also $T_2$ satisfies $\forall a$. Now, continue to define $T_i$ for all $i \geq 0$ in the same way, so that in each $T_i$ there are exactly $i$ blanks (non-marked vertices), and $v_1, v_2, \ldots, v_i$ are decreasing with respect to $\forall a$.

Now, define $T_\infty$ to be the tree having the same nodes as all the $T_i$, with all of them marked $a$, except the $u_i$, which are left blank. $T_\infty$ has the following form:

$$T_\infty: 0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0\quad--\quad0$$

By the definition of $T_\infty$ we have that each $v_i$ is decreasing with respect to $\forall a$, and hence, by lemma 3, $T_\infty$ satisfies $\forall a$, which is not true, since it has infinitely many blanks (all the $u_j$) - a contradiction. This completes the proof of theorem 3.

Remark: $T_\infty$ is a recursive tree since $R$ is recursive. If we do not restrict the discussion
to recursive trees only, then the recursiveness assumption on $R$ is superfluous.

Although alternative proof rule 1 is impossible, one might think of another possibility, which still seems more natural than the suggested rule. Up to now, there has always been a predicate, which tells, given a vertex in a marked tree, whether or not its successors should decrease in rank. In case of a negative answer, its successors should have remained steady. However, it is also possible to allow increment of the rank, i.e., instead of a predicate, we could have a function $f_{\Sigma,\phi}(e^T) \rightarrow \{d, s, i\}$, where $d, s$ and $i$ mean decrease, steady and increase respectively, and then suggest the following proof rule:

**Alternative proof rule 2:** To prove $\text{avoid}(T, \phi)$, find a well founded set $(W, <)$ and a predicate $P: W \times T \rightarrow \{T, F\}$ s.t the following conditions hold:

1. There exists $w \in W$ such that $P(w, \text{root})$ holds.

2. For all $v \in T$ and for all $w \in W$,

   (a) $P(w, v) \land f_{\phi}(\pi_v) = d \rightarrow \forall s \in \text{children}(v) \exists w' < w: P(w', s)$.

   (b) $P(w, v) \land f_{\phi}(\pi_v) = s \rightarrow \forall s \in \text{children}(v) \exists w' \leq w: P(w', s)$.

   (c) $P(w, v) \land f_{\phi}(\pi_v) = i \rightarrow \forall s \in \text{children}(v) \exists w' > w: P(w', s)$.

In this rule, if $f$ returns $i$ for some node, then its successors can have any ranks, including higher ones. At first sight, this rule seems to work perfectly, since the "problematic" formula $\lor_a$ is no longer such if we define:

$$f\lor_a(\pi) = \begin{cases} 
  d & \text{if } a \text{ appears in the last element of } \pi \\
  i & \text{else}
\end{cases}$$

It can be easily verified, that alternative rule 2 is sound and complete for proving $\text{avoid}(T, \lor_a)$, if we adhere to the suggested $f$. We could also define $f$ for all the other atomic formulas, in a way similar to the original proof rule (this is immediate, since in all the atomic formulas except $\lor_a$, the two arguments $\pi_1$ and $\pi_2$ are concatenated,
and so we relate to the fragment from the root to the vertex only, and so we do not need two arguments). It therefore turns out, that one could come with a sound and complete proof rule of the last form for each atomic formula. It also turns out, that it is possible to define \( f \) for a conjunction of two formulas. The problem arises when we try to bring in disjunctions. We now prove that it is impossible to find such an \( f \) to solve the general case.

**Theorem 4:** There exists no recursive \( f \) with which alternative proof rule 2 is sound and complete.

**Proof:** By way of contradiction, let \( f \) be such a function. We refer to nodes \( v \), for which \( f,\varphi(v) = d \) as decreasing with respect to \( \varphi \), and steady and increasing is used similarly. If \( \varphi \) is understood from the context we omit the "with respect to \( \varphi \)". We need two lemmas for the proof.

**Lemma 4:** Let \( T \) be a tree composed from one path only, \( v_1, v_2, v_3, \ldots \), and let \( \varphi \) be a formula. If \( T \) satisfies \( \varphi \) then there are infinitely many decreasing nodes in \( T \).

**Proof (of lemma 4):** Similar to the proof of lemma 3. The fact that ranks can increase now is immaterial.

\[ \square \]

**Lemma 5:** Let \( T \) and \( \varphi \) be as in lemma 4. If there are infinitely many increasing nodes in \( T \), then avoid\((T,\varphi)\) holds.

**Proof (of lemma 5):** Assume there are infinitely many increasing nodes in \( T \). For each \( k \geq 1 \) let \( n_k \geq 1 \) be the minimal integer s.t. \( v_{k+n_k} \) is a increasing vertex. i.e, for a vertex \( v_k \), \( v_{k+n_k} \) is its first increasing descendent. Since there are infinitely many increasing nodes, \( n_k \) is defined for all \( k \). Now, define \( W=(N,\leq) \) and a predicate \( P:W \times T \rightarrow \{T,F\} \) as follows:

\[ P(w,v_k)=T \iff (k=1 \text{ and } w=n_1+1) \text{ or } (k \geq 2 \text{ and } w=n_{k-1}). \]

It is easy to see that conditions (1) and (2) hold with \( W \) and \( P \), and so, by the soundness
of the rule, \( \text{avoid}(T, \varphi) \) holds.

We now proceed with the proof of theorem 4. Let \( T_0 \) be as in the proof of theorem 1.

Let \( T_0 \) be as in the proof of theorem 1.

Thus, there must be infinitely many increasing nodes in \( T_0 \) or else we would conclude, by a slight variation of lemma 3, that \( T_0 \) satisfies \( (\forall \alpha \exists \beta) \), which is untrue. Let \( \nu_1 \) be the first increasing vertex, and let \( \nu_1 \) be its successor. \( T_1 \) is the following tree:
i.e. all nodes up to, and including \( v_1 \) are marked as in \( T_{0,\ast} \) (their marks are represented by an \( ? \)), \( u_1 \) is marked \( b \), and all other nodes are marked \( a \). \( v_1 \) is increasing with respect to \( (\forall^a_a \lor \exists^b_b) \). Now, since \( T_1 \) satisfies \( (\forall^a_a \lor \exists^b_b) \), there are, by lemma 4, infinitely many decreasing nodes in it. Let \( v_{1,1} \) be the first decreasing node after \( u_1 \), and let \( u_{1,1} \) be its successor. \( T_{1,1} \) is the following tree:

\[
\begin{array}{cccccccccccc}
\vdots & ? & ? & ? & b & a & a & a & a & a & a & a & \vdots \\
\vdots & v_1 & u_1 & v_{1,1} & u_{1,1} & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{array}
\]

i.e. \( u_{1,1} \) is blanked, and all nodes following it are marked \( a \). All nodes up to \( v_{1,1} \) are marked as in \( T_1 \). Again, \( v_{1,1} \) is decreasing, and \( T_{1,1} \) satisfies \( (\forall^a_a \lor \exists^b_b) \). Continue and define \( T_{1,i} \) for all \( i \) analogously to the definition of \( T_{0,i} \) before, so that each \( T_{1,i} \) has exactly \( i \) blanks after \( u_1 \) (all the \( u_{1,i} \)), and \( v_{1,1} \ldots v_{1,i} \) are decreasing.

Now, define \( T_{1,\ast} \) from the \( T_{1,i} \) analogously to the definition of \( T_{0,\ast} \) from the \( T_{0,i} \). In \( T_{1,\ast} \) there are infinitely many decreasing nodes, and thus by an argument already used once, there are infinitely many increasing ones. Let \( v_2 \) be the first increasing vertex after \( u_1 \), and let \( u_2 \) be its successor. Define \( T_2 \) as follows:

\[
\begin{array}{cccccccccccc}
\vdots & v_1 & u_1 & v_2 & u_2 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{array}
\]

i.e. all nodes up to, and including \( v_2 \) are marked as in \( T_{1,\ast} \); \( u_2 \) is marked \( b \), and all other nodes are marked \( a \). Note that in \( T_2 \) there are exactly two nodes marked \( b \) (\( u_1, u_2 \)).

Continue and define \( T_i \) for all \( i \) in the same way. Each \( T_i \) has exactly \( i \) nodes marked \( b \) (\( u_1 \ldots u_i \)), and has at least \( i \) increasing nodes (\( v_1 \ldots v_i \)). Define \( T_{\ast} \) as the tree having the same nodes as all the \( T_i \), and marks which are obtained from them in the usual way (more formally, a vertex \( v \) is marked as in \( T_k \), where \( k \) is the minimal integer \( s.t. \) \( u_k \) is a descendent of \( v \)). In \( T_{\ast} \) there are infinitely many increasing nodes (the \( u_i \)), and so by lemma 5, \( \text{avoid}(T_{\ast}, (\forall^a_a \lor \exists^b_b)) \) holds, but then, there are infinitely many nodes marked \( b \), which leads to the satisfaction of \( (\forall^a_a \lor \exists^b_b) \) - a contradiction.
Remark: Again, $T_\omega$ is recursive since $f$ is such, and this assumption on $f$ can be omitted if we do not restrict the discussion to recursive trees.

6. Conclusion

We have presented a necessary and sufficient condition for a marked tree to have no infinite paths satisfying a given formula, taken from a rather expressive language. Two other simpler forms of conditions were also discussed and were shown to be impossible. The completeness proof is a generalization of the "cone construction" [GFMR 81, F 85].

Considering the case of fairness, we observe that the existing proof rules for termination under this assumption are close in spirit to the simpler conditions, those which were shown here to be inadequate. They are close in the sense that only one well founded set has to be found in order to complete the proof, rather than one for each subtree, as it is here. The ability to use a simpler rule in the fairness case stems from the fact, that it is a very special case of the general one, both in the markings allowed and in the formula used (see [Ha 84] for the translation of fairness to trees and formulas).

As mentioned before, we omitted the discussion of infinite conjunctions and disjunctions. It, therefore, remains to close this gap. We conjecture that the kind of condition discussed here cannot be applied to the infinite formulas of $L$; We have, however, no proof of this. One natural attempt yields a nonrecursive tree as a counter example.

A most natural extension of the work is to apply the condition to programs, i.e to provide a syntactic proof rule for proving that a program has no infinite computations satisfying a given formula from $L$. A first stage of such a work would be to properly interpret markings and formulas when relating to programs.

Another intriguing direction of research could be characterizing the $\&$-avoidance of a marked tree in terms of convergent sequences of computation elements in
appropriate metric spaces. This approach is discussed in [DM 84] for the case of fairness and three other liveness properties. Relating to fairness, it is shown there how to define a distance between computation elements of a program in such a way, that a given computation is fair iff it is a Cauchy sequence. This approach is further developed in [C 84], where Milner's CCS is used as the computational model. It would be of interest to define, for each \( \varphi \in L \) a distance function \( d_\varphi \) in such a way, that a path in the tree satisfies \( \varphi \) iff it forms a Cauchy sequence (with respect to \( d_\varphi \)).

Acknowledgments

We acknowledge J.A. Makowsky's remarks, suggesting the need for this kind of investigation.

The part of the second author was partially supported by the fund for the promotion of research, the Technion.

References

[DM 84] N. Dror, D. Makowsky
Salt Lake City 1984

[FK 84] N. Frascarelli

[CFK 83] O. Czerwinski
theor. 1983

[GFMR 81] O. Grumberg

[KA 84] D. Harms
ACM-STOC, May 1984
REFERENCES


