INFORMATION AND COMPUTATION **124**, 127–144 (1996)
ARTICLE NO. 0010

A Temporal Logic for Proving Properties of Topologically General Executions

RACHEL BEN-ELIYAHU*

Computer Science Department, Technion—Israel Institute of Technology, Haifa, Israel 32000

AND

MENACHEM MAGIDOR

Mathematics and Computer Science Institute, The Hebrew University, Jerusalem, Israel 91904

We present a generalization of the temporal propositional logic of linear time which is useful for stating and proving properties of the generic execution sequence of a parallel program or a non-deterministic program. The formal system we present is exactly that same as the third of three logics presented by Lehmann and Shelah (*Information and Control* **53**, 165–198 (1982)), but we give it a different semantics. The models are tree models of arbitrary size similar to those used in branching time temporal logic. The formulation we use allows us to state properties of the "co-meagre" family of paths, where the term "co-meagre" refers to a set whose complement is of the first category in Baire's classification looking at the set of paths in the model as a metric space. Our system is decidable, sound, and, complete for models of arbitrary size, but it has the finite model property; namely, every sentence having a model has a finite model. © 1996 Academic Press, Inc.

1. INTRODUCTION

There is an extensive literature dealing with correctness and termination arguments for non-deterministic or distributed processes in which the required condition cannot be guaranteed to hold for all possible executions. Rather, one can only guarantee correctness for a "general," "generic," or "not-exceptional" execution. For example, in the study of many probabilistic algorithms (e.g., [CLP83, LR81]), one is only able to show that the desirable behavior holds with high probability (i.e., 1).

There is a vast literature on "fairness" conditions (e.g., [LPS81, Pnu83, Fra86] where one restricts one's attention to only "fair" executions, with different possible definitions of fairness. Note that in the motivation for this study there is an implicit assumption that the "fair" execution sequences are most of the execution sequences, that the "general" sequence is fair (otherwise it seems necessary to make the algorithm explicitly care about fairness, but then every execution is fair, and then the study of fairness assumptions seems meaningless).

There are many inequivalent definitions of fairness (see [Fra86]) and there are examples of algorithms which are correct for one notion of fairness but not correct for a weaker notion (see for example [LPS81]). The common theme running through all the definitions of fair executions is the requirement that the algorithm will be correct in the general case. This common theme is the one investigated in this paper.

Lehmann and Shelah [LS82] introduced a generalization of the temporal propositional logic, which can be used for stating and proving properties of probabilistic programs, when the properties proved are true for most executions (in the probabilistic sense). Their models are stochastic systems, with state transition probabilities. They have presented three different decidable axiomatic systems, and showed that they are sound and complete for generally models, finite models, and models with bounded transition probabilities, respectively. Their language includes, in addition to the connectives usually used in temporal logic, a new connective denoted by ∇ and called "certainly." If a is a formula, the formula ∇a is true iff a is true with probability 1.

In reading [LS82] one is struck by the fact that there is a vast freedom in choosing the probability distribution in the model constructed in its completeness proof. Namely, "most" executions will have the required properties with almost any reasonable probability distribution. So it seems that there is an underlying notion of a general set of executions which is more basic then having probability 1 with respect to a particular distribution.

These are the considerations that led to this work. In this paper we try to formalize the notion of "general execution." We feel that an adequate formalization calls for considering the topology of the space of possible execution sequences,

[*] Most of this work is the master's thesis of the first author, completed at the Hebrew University, Jeruzalem Israel. Part of this work was done while the first author was a student at the Cognitive Systems Laboratory, Computer Science Department, University of California, Los Angeles, California.

0890-5401/96 $18.00

Copyright © 1996 by Academic Press, Inc.
All rights of reproduction in any form reserved.

and that the notion of "meagre set" well known to topologists is an adequate formalization of "exceptional," "non-general," etc.

Therefore, we reinterpret the extension of temporal logic suggested by [LS82] by interpreting the connective ∇ as "for a co-meagre set of paths," which intuitively means "generally it is true that...." Supporting evidence for this formalism is the fact that for any reasonable notion of "general," "generic," "fair" introduced in the literature (see for example [Fra86]), the corresponding set of execution sequences turns out to be a co-meagre set in the appropriate topology.

It turns out that the system introduced by [LS82] for finite probabilistic models is sound and complete for this topological interpretation, even when one considers models of arbitrary size. Another very pleasing property of our suggested formalization of the notion of "general execution" is the fact that follows from Theorem 8.2 that if we consider a general execution sequence, namely one that avoids any set of first category that can be described in our language, then every property which is definable in our language has a finite character; namely, if a general sequence has that property, then there is a finite initial segment of it such that every general sequence extending this initial segment has the property.

In [LPS81] a proof system is presented for proving termination of fair executions for different definitions of fairness. In that system each proof assumes some underlying well founded relation, for which one can prove that under certain circumstances we get a value which is smaller under this relation. Supplying such a relation may be difficult in practice.

Our approach works only for programs such that their behavior could be expressed using propositional logic. This includes many programs such as operating systems where the number of values that can be written in each register is bounded, etc., but of course this is not the most general case.

In [AS85] the authors considered different properties of execution sequences in a topological context. They showed that what they called "liveliness" properties are dense in the space of possible executions. We feel that the fact that a certain set of execution sequences is dense is not a sufficient argument that a general execution will belong to this set (for example, the set can be a dense countable set). Dense sets can be made of rather specific and nontypical execution sequences, and therefore the behavior on a small dense set does not gives the behavior on "most" or "general" execution. For instance, the set of non-fair executions can also be dense. One needs further arguments; for example, if the set is of "finite character" (namely, having the required property is determined at a finite stage), only then is being dense the same as being co-meagre.

The completeness proof is in principle like the proof of [LS82l for finite models, and we would have liked simply to refer the reader as much as possible to this proof. However, the proof for the systems TCB and TCF contains a subtle mistake, and in order to avoid this mistake we must reverse the proof, redefining one of the key concepts. The proof we give suggests how to fix that completeness proof. The mistake and a counterexample to one of the key lemmas of [LS82] appear in the Appendix. We still use as much as possible the proof of [LS82]. For readers' convenience we repeat all the needed definitions and quote the lemmas which we can adopt verbatim.

The rest of the paper is organized as follows: In the next section we give basic definitions and lemmas from set topology to establish a terminology that we will use later. In Section 3 we define the model of the logical system, and show how the notion of fair executions translates to the concept of a co-meagre set. Then we present the language, the semantics, the definitions of satisfiability and validity, and the set of axioms and inference rules in Sections 4 to 7, respectively. In Section 8 we show that the logical system is sound, and in Section 9—that it is complete. In Section 10 we present two examples of the use of our system to prove desirable properties (such as termination, mutual exclusion, and freedom from starvation) of parallel programs, and in Section 11 we provide concluding remarks.

2. SOME BASIC RESULTS FROM SET TOPOLOGY

This section contains some basic results from set topology that we will use later. Some of the proofs are given in [BE87], while the others are obvious or can be found easily in any introductory book in tology (see for example [Ke55] or [Dix84]).

We will use (X, d) or just X to denote a metric space X with the distance function d, and A, B, C, D... to denote subsets of X. $\mathcal{N}$ will denote the set of nonnegative integers, and a nonnegative integer will be denoted by i, j, k...

2.1. Categories of Sets

We remind the reader that a subset A of (X, d) is *nowhere dense* (notation: nowd(X)) if $\forall B \subseteq X$ such that B is *open* in the metric space X (notation: open(X)) and $B \neq \emptyset$, $\exists B' \subseteq B$ such that $B' \neq \emptyset$, B' is open(X) and $A \cap B' = \emptyset$.

Definition 2.1. A subset of X is called "*of first category*" (notation: fc(X)) if it can be represented as a union of countably many nowhere dense sets.

Note that sets of first category can still be dense, for example, the set of all rational numbers is dense but is of first category.

Definition 2.2. A subset of X is called "*of second category*" (notation: sc(X)) if it is not of first category.

Note that if A is sc(X), A is not empty, because the empty set is fc(X).

DEFINITION 2.3. A subset of X is called "*co-meagre*" (notation: co-m(X)) if its complement in X is of first category.

It follows from the definition that if X is a metric space, X is co-m(X).

LEMMA 2.4. *If A is* co-m(X) *and B is* sc(X) *than $A \cap B$ is* sc(X).

LEMMA 2.5. *Let $\mathscr{A}$ be an arbitrary size set of first category sets. If for every member A_i of $\mathscr{A}$ there is an open set G_i such that $A_i \subseteq G_i$, and for every $i \neq j$ G_i and G_j are disjoint, then the union over $\mathscr{A}$ is of first category.*

LEMMA 2.6. *Let $A \subseteq X$. If for every set B such that $B \neq \varnothing$ and B is* open(X) *there is an open set C such that $C \neq \varnothing$, $C \subseteq B$ and $A \cap C$ is* fc(X), *then A is* fc(X).

2.2. The Interior and Closure of a Set

DEFINTION 2.7. The *interior* of a set A (notation: Int A) is the maximal open set B such that $B \subseteq A$.

DEFINITION 2.8. The *closure* of A (notation: $\bar{A}$) is the minimal closed set B such that $A \subseteq B$.

LEMMA 2.9. *F is* nowd(X) *iff* Int $\bar{F} = \varnothing$.

2.3. Almost-Open Sets

DEFINITION 2.10. $A \subseteq X$ is *almost-open*[1] (notation: almost-open(X)) if there exist a subset B of X such that B is open(X) and $A \triangle B$ is fc(X).[2]

An almost open set is known in the literature as "a set which has Baire's property".

LEMMA 2.11. *F is* almost-open(X) *iff its complement is* almost-open(X).

LEMMA 2.12. *If A is* open(X) *and B is* almost-open(A), *then B is also* almost-open(X).

LEMMA 2.13. *A union over a countable set of almost-open sets is almost-open. The intersection of finitely many almost open sets is almost open.*

LEMMA 2.14. *Let H be an arbitrary size set of almost open sets. So for every $H_i \in H$ there are sets G_i, A_i, B_i such that G_i is* open(X), *B_i and A_i are* fc(X), *and $H_i = (G_i \cup A_i) - B_i$. If for every $H_i \in H$ there is an open set E_i such that $A_i \subseteq E_i$, and an open set F_i such that $B_i \subseteq F_i$ and for each $i \neq j$ $E_i \cap E_j = \varnothing$ and $F_i \cap F_j = \varnothing$, then $\bigcup_{H_i \in H} H_i$ is* almost-open(X).

2.4. A Complete Metric Space

DEFINITION 2.15. A metric space X is *complete* if every Cauchy sequence $\{p_i \mid i = 1, 2, ...\}$ of points converges.

THEOREM 2.16. (Baire's Theorem). *Let X be a complete metric space, and let $A \subseteq X$. If A is* fc(X) *then* Int $A = \varnothing$.

COROLLARY 2.17. *If X is a complete metric space, then X is* sc(X).

LEMMA 2.18. *Let X be a complete metric space, and let $A \subseteq X$. If A is* co-m(X) *then A is* sc(X).

2.5. Homeomorphism between Metric Spaces

DEFINITION 2.19. A function f from a metric space X onto a metric space Y is *continuous* iff for every G which is open(Y), $f^{-1}(G)$ is open(X).

DEFINITION 2.20. A continuous function f from a metric space X onto a metric space Y is called a *homeomorphism* iff f is one-to-one and f^{-1} is a continuous mapping from Y onto X. If there is a homeomorphism from X onto Y we call X and Y *homeomorphic*.

3. THE MODEL

3.1. The Model and Its Topological Features

We suppose that a set *Pvar* of propositional variables is given. 2^{Pvar} will denote the set of all subsets of *Pvar*.

DEFINITION 3.1. A model M is a quadruple $\langle S, u, l, R \rangle$ where the following holds:

- S is an arbitrary nonenmpty set. Elements of S are called states and denoted by s, t, u...
- $u \in S$ is called the initial state.
- l: $S \rightarrow 2^{Pvar}$ is a labeling function associating to every state the set of propositional variables that hold in that state.
- $R \subseteq S \times S$ a serial relation; i.e., for each $s \in S$, there exists a state t such that $R(s, t)$.

Note that a state s in a model may have more than one R-successor.

DEFINITION 3.2. A path in a model M is an infinite sequence of states $s_0, s_1, s_2, ...$

Paths will be denoted by ρ, σ, τ... We shall denote the nth state of a path σ by $\sigma(n)$, and we shall use σ^{+n} to denote the path defined by $\sigma^{+n}(m) = \sigma(n+m)$.

[1] An equivalent definition for almost-open: A is almost-open(X) if there are sets B, C, D such that B is open(X), C and D are fc(X), and $A = (B \cup C) - D$.

[2] $A \triangle B$ is the symmetric difference between A and B, that is, $A \triangle B = A \cup B - A \cap B$.

A path σ is *legal* iff $\forall i \in \mathcal{N}$, $R(\sigma(i), \sigma(i+1))$. From now on when we use the term "path" we mean a legal path.

For a given state s, we will denote by P_s the set $\{\sigma \mid \sigma$ is a legal path, $\sigma(0)=s\}$. Note that for every $s \in S$ $P_s \neq \emptyset$ because R is serial.

For the logic we consider, every model corresponds to a tree model which satisfies the same set of formulas. Hence we will view the model as an infinite tree, where each node in the tree is labeled by a state, where a state is actually a subset of *Pvar* which hold in this state, the root is the initial state u, and the children of each node s are all and only the members of the set $\{t \mid t \in S, R(s, t)\}$. So a path in a model is actually a path in the tree which represents the model.

Definition 3.3. Let $M = \langle S, u, l, R \rangle$ be a model, and let $s \in S$. We shall move P_s into a metric space by defining the distance d between any 2 paths $\sigma_1, \sigma_2 \in P_s$, denoted by $d(\sigma_1, \sigma_2)$, as follows: if $\sigma_1 = \sigma_2$ then $d(\sigma_1, \sigma_2) = 0$; else $d(\sigma_1, \sigma_2) = 1/n$, where $n \in \mathcal{N}$ satisfies the following conditions: $\sigma_1(n) \neq \sigma_2(n)$ and $\forall m \in \mathcal{N}$, $m < n \Rightarrow \sigma_1(m) = \sigma_2(m)$.

Note that always $n \neq 0$ (because we define the distance only between paths that start with the same state), and that the distance between two paths is always between 0 and 1.

The idea of viewing a set of executions of a concurrent program as a metric space appears already in [AN80, BZ82], in the context of *semantics* of concurrent programs.

Let $M = \langle S, u, l, R \rangle$ be a model. It can be easily shown that for every $s \in S$, (P_s, d) is a *complete* metric space.

Definition 3.4. Let $s \in S$, $n \in \mathcal{N}$. The *n-environment* of a path $\sigma \in P_s$ (notation: n-env(σ)) is defined as follows: if $n > 0$, then n-env$(\sigma) = \{\sigma' \mid \sigma' \in P_s, d(\sigma, \sigma') < 1/n\}$. if $n = 0$, then n-env$(\sigma) = P_s$.

The set n-env(σ) is the set of paths that coincide with σ in the first $n+1$ states.

Lemma 3.5. *Let $M = \langle S, u, l, R \rangle$ be a model. Let $s \in S$, $\tau \in P_s$, $n \in \mathcal{N}$. The sets* n-env(τ) *and $P_{\tau(n)}$ are homeomorphic.*

Definition 3.6. The homeomorphism described in Lemma 3.5 will be called the *natural homeomorohism.*

Lemma 3.7. *Let $\langle S, u, l, R \rangle$ be a model. Let $s \in S$, $\tau \in P_s$, $n \in \mathcal{N}$. If a set G is* open$(P_{\tau(n)})$ *then the set $\{\sigma \mid \sigma \in P_s, \sigma^{+n} \in G, \sigma \in n$-env$(\tau)\}$ is* open(P_s).

3.2. Fairness in the Model

In this section we will show that for some known definitions given in the literature for fairness, the set of unfair executions is of first category.

First, we must explain how a concurrent program can be represented in the model. Each state of the model represents a state of the program—the values of all the program variables, the values of the program counters of all its processes, etc. The initial state represents the initial state of the program. For states s, t we will have $R(s, t)$ if there is a process that can transfer the program from a state s to a state t in one atomic action. So each path in the model represents a possible execution sequence of the program and therefore we will often refer to a path as an execution sequence. A terminating state will be a state where the program can longer move to a different state.

Since each path in the model is infinite, a finite execution sequence will be a path in which only the terminating state appears from some point on, i.e., a path σ denotes a finite execution if $\exists n \in \mathcal{N}$ such that v is a terminating state and $\forall m \geqslant n$ $\sigma(m) = v$. If a path σ denotes a finite execution we will call it a converging path.

Let F be the set of all the processes of the program. Our model is not restricted to a countable set of processes, because S can be of any cardinality, but in this discussion we will assume that F is a countable set.

For a path σ and a process f, we will say that f appears in σ in step n if $\sigma(n) = s$, $\sigma(n+1) = t$, and the process f can be activated at state s of the program and move it to state t. A process f appears k times in σ if there is $A \subset \mathcal{N}$ such that $|A| = k$ and $\forall i \in A$ f appears in σ in step i. A process f is enabled in σ in step n if $\exists t$ such that the process f can transfer the program from the state $\sigma(n)$ to the state t.

Each state in our model can be regarded as a model for propositional logic in an obvious way. A formula in propositional logic will be called a *property*, and will be denoted by ϕ or ψ. We will say that a property ϕ occurs in σ if there is a state along σ that satisfies ϕ, , i.e., if $\exists n \in \mathcal{N}$ such that $\sigma(n)$ satisfies ϕ.

Lehmann *et al.* [LPS81] consider three types of fairness: Impartiality, Justice, and Fairness.

- *Impartiality.* A path is defined to be impartial if it is either converging or such that $\forall f \in F$, f appears infinitely many times in the path.

Of course here we assume that if the program is not finite, every process is enabled an infinite number of times in any execution sequence.

- *Justice.* A path is defined to be just if it is converging or if every r process which is continuously enabled beyond a certain point appears in the path infinitely many times.

- *Fairness.* A path is said to be fair if it is converging or if every process that is enabled an infinitely many times appears in the path infinitely many times.

Let $M = \langle S, u, l, R \rangle$ be a model. We will show that the set of unfair executions is of first category. Since injustice or not impartial executions are only a special case of unfair executions we will get as a consequence that the set of impartial executions and the set of unjust executions are also of first category.

LEMMA 3.8. *Let $M = \langle S, u, l, R \rangle$ be a model which represents a program. The set of unfair executions of the program is* $\mathrm{fc}(P_u)$.

Proof. For every $f \in F$ we will show that the set $A_f =_{\mathrm{def}} \{\sigma \mid \sigma \in P_u, f$ is enabled an infinite number of times in σ but appears only a finite number of times in $\sigma\}$ is $\mathrm{fc}(P_u)$.

Let $f \in F$, $n \in \mathcal{N}$. Let $B_{f,n} =_{\mathrm{def}} \{\sigma \mid \sigma$ is not converging, f is enabled an infinite number of times in σ, f appears in σ exactly n times$\}$. Clearly, $\bigcup_{n \in \mathcal{N}} B_{f,n} = A_f$. We will show that $B_{f,n}$ is nowhere dense.

Let G be $\mathrm{open}(P_u)$ and suppose $G \cap B_{f,n} \neq \emptyset$. Let $\tau \in G \cap B_{f,n}$. Let i be the index such that f does not appear in τ after step i. Since G is open, $\exists k$ such that the k-environment of τ is in G. Let $m = \mathrm{MAX}\{i, k\}$. Since f is enabled an infinite number of times in τ, $\exists j$ such that $j > m$ and f is enabled in step $\tau(j)$. Since f is enabled in step j of τ, $\exists t \in S$ such that $R(\tau(j), t)$.

Let σ be the path such that $\forall i \; 0 \leqslant i \leqslant j \; \sigma(i) = \tau(i)$, $\sigma(j+1) = t$. Clearly, $E =_{\mathrm{def}} \{$The $j+1$ environment of $\sigma\}$ is $\mathrm{open}(P_u)$, $E \subseteq G$. Since for every $\sigma \in E \cap B_{f,n}$ f appears in σ at least $n+1$ times, $E \cap B_{f,n} = \emptyset$. ▮

Francez [Fra86] gives a general notion of fairness. He defines a fairness condition **F** as a finite, non-empty set of pairs of state properties, $\mathbf{F} = \{(\phi_j, \psi_j) \mid 1 \leqslant j \leqslant K\}$, where K is a natural number. Francez then suggests three types of generalized fairness, where σ is an execution sequence and **F** a fairness condition:

- *Unconditional* **F**-*fairness.* σ is unconditionally **F**-fair iff for each j, $1 \leqslant j \leqslant K$, ψ_j occurs infinitely often along σ.
- *Weak* **F**-*fairness.* σ is weakly **F**-fair iff for each j, $1 \leqslant j \leqslant K$: if ϕ_j occurs continuously along σ, then ψ_j also occurs infinitely often along σ.
- *Strong* **F**-*fairness.* σ is strongly **F**-fair iff for each j, $1 \leqslant j \leqslant K$: if ϕ_j occurs infinitely often along σ, so does ψ_j.

We call a fairness condition $\mathbf{F} = \{(\phi_j, \psi_j) \mid 1 \leqslant j \leqslant K\}$ *feasible* iff whenever ϕ_j holds in a state s there is a set of processes $p_1, ..., p_n$ and a set of states $s_0, ..., s_n$ such that $s_0 = s$, for every $1 \leqslant i \leqslant n$ the process p_i can transfer the program from state s_{i-1} to state s_i and ψ_j holds in s_n. From now on we restrict our attention to *feasible* fairness conditions (we believe that these are the interesting cases). A more general definition of a feasible fairness condition can be found in [AFK88].

We also assume that in unconditional **F**-fairness, if the path is not converging, ϕ_j holds infinitely many times along σ.

We can show that under the above assumptions, the set of strongly fair paths is co-meagre. The proof is very similar to the proof of Lemma 3.8 above, and since the other types of fairness is only a special case of this one, we get that also according to Francez' definitions (with reasonable assumptions), the set of all fair executions is co-meagre.

4. THE LANGUAGE

Our language is the same language as presented by [LS82]. The formulas are composed from propositional variables, classical connectives, temporal connectives, and modal connectives. In the formal definition that follows, the set of all the formulas, Γ, will be defined, together with the "size" ($\#$) of a formula. Propositional variables will be denoted by p, q, ..., formulas by a, b...

DEFINITION 4.1. The set Γ of all the formulas is defined by the following rules:

1. A propositional variable $p \in Pvar$ is a formula and $\#(p) = 1$. A propositional variable denotes a basic proposition, that does not mention time.

2. If b and c are formulas, then:

- $\neg b$ is a formula and $\#(\neg b) = \# b) + 1$.
- $b \vee c$ is a formula and $\#(b \vee c) = \#(b) + \#(c) + 1$.
- $\bigcirc b$ is a formula and $\#(\bigcirc b) = \#(b) + 1$. The symbol $\bigcirc$ is read "next" and denotes the next instant of time.
- $\Box b$ is a formula and $\#(\Box b) = \#(b) + 1$. The symbol $\Box$ is read "always" and denotes all the instants of time from the present (included) and on.
- b **Until** c is a formula and $\# b$ **Until** $c = \#(b) + \#(c) + 4$ (we need $\#(b$ **Until** $c) > \#(\neg(b \vee c))$ for the completeness proof). The symbol **Until** is read "until." The formula b **Until** c denotes the fact that, there is an instant of time in the future when c is true and until the first such instant of time, b stays continuously true at all intermediate instants of time.
- ∇b is a formula and $\#(\nabla b) = \#(b) + 1$. The symbol ∇ is read "Generally" and denotes "for almost all" the paths that begin from the present state.

We use the following abbreviations: $b \wedge c$ for $\neg(\neg b \vee \neg c)$, **true** for $p \vee \neg p$, **false** for $\neg$**true**, $b \rightarrow c$ for $\neg b \vee c$ and $b \leftrightarrow c$ for $(b \rightarrow c) \wedge (c \rightarrow b)$. Other abbreviations are as follows: $\Diamond b$ is read "sometimes b" and stands for $\neg \Box \neg b$, $\triangle b$ is read "possibly b" and stands for $\neg \nabla \neg b$. The rules of precedence are as usual, and $\rightarrow$ associates to the right.

5. THE SEMANTICS

In this section we will show how we assign a truth value to every formula in every path of the model.

Definition 5.1. Let M be a model $\langle S, u, l, R\rangle$, σ a path in M, and $a \in \Gamma$ a formula. Then

$$p \mid_M^\sigma = \mathbf{true} \Leftrightarrow p \in l(\sigma(0)).$$

That means that the truth value of a propositional variable relative to (or in) a path depends only on the first state of the path. Further,

$$\neg a \mid_M^\sigma = \mathbf{true} \Leftrightarrow a \mid_M^\sigma = \mathbf{false}$$

$$a \vee b \mid_M^\sigma = \mathbf{true} \Leftrightarrow a \mid_M^\sigma = \mathbf{true} \text{ or } b \mid_M^\sigma = \mathbf{true}$$

$$\bigcirc a \mid_M^\sigma = \mathbf{true} \Leftrightarrow a \mid_M^{\sigma^{+1}} = \mathbf{true}$$

$$\Box a \mid_M^s = \mathbf{true} \Leftrightarrow \forall n \in \mathcal{N}\ a \mid_M^{\sigma^{+n}} = \mathbf{true}$$

$$a \ \mathbf{Until}\ b \mid_M^\sigma = \mathbf{true} \Leftrightarrow \exists n \in \mathcal{N} \text{ such that } b \mid_M^{\sigma^{+n}} = \mathbf{true}$$
$$\text{and } \forall k < n,\ a \mid_M^{\sigma^{+k}} = \mathbf{true}$$

$$\nabla a \mid_M^\sigma = \mathbf{true} \Leftrightarrow \text{the set } \{\sigma' \mid \sigma' \in P_{\sigma(0)},\ a \mid_M^{\sigma'} = \mathbf{true}\}$$
$$\text{is co-m}(P_{\sigma(0)}).$$

Note that the truth of a formula of the type ∇a at a path σ in a model M depends only on the state $\sigma(0)$ and the model M.

6. SATISFACTION AND VALIDITY

Intuitively, our notion of satisfiability says that a model satisfies a formula a if a holds for *almost all* the paths beginning at the initial state. The intuition behind the definition of satisfiability in [LS82] is similar, but in their semantics "*almost all* paths" means a set of paths that have probability one. Here is our formal definition for satisfiability:

Definition 6.1. Let $M = \langle S, u, l, R\rangle$ be a model and $a \in \Gamma$ a formula. We say that M satisfies a and write $M \models a$ if the set $\{\sigma \mid \sigma \in P_u,\ a \mid_M^\sigma = \mathbf{true}\}$ is co-m(P_u).

Lemma 6.2. *Let $M = \langle S, u, l, R\rangle$ be a model. $M \models a \Leftrightarrow M \models \nabla a$.*

Proof.

$\Rightarrow$: $M \models a \Rightarrow \{\sigma \mid \sigma \in P_u,\ a \mid_M^\sigma = \mathbf{true}\}$ is co-m$(P_u) \Rightarrow$ $(\forall \sigma, \sigma \in P_u \Rightarrow \nabla a \mid_M^\sigma = \mathbf{true}) \Rightarrow \{\sigma \mid \sigma \in P_u, \nabla a \mid_M^\sigma = \mathbf{true}\} = P_u$ is co-m(P_u), $\Rightarrow M \models \nabla a$.

$\Leftarrow$: $M \models \nabla a \Rightarrow A =_{\text{def}} \{\sigma \mid \sigma \in P_u, \nabla a \mid_M^\sigma = \mathbf{true}\}$ is co-m $(P_u) \Rightarrow A$ is sc(P_u) (see Lemma 2.18) $\Rightarrow A$ is not empty $\Rightarrow \exists \sigma \in P_u, \nabla a \mid_M^\sigma = \mathbf{true} \Rightarrow \{\sigma \mid \sigma \in P_u,\ a \mid_M^\sigma = \mathbf{true}\}$ is co-m $(P_u) \Rightarrow M \models a$. ∎

Note that in any model $M = \langle S, u, l, R\rangle$ it may happen that both $M \not\models a$ and $M \not\models \neg a$. That will happen when the set $\{\sigma \mid \sigma \in P_u,\ a \mid_M^\sigma = \mathbf{true}\}$ is sc(P_u), and the set $\{\sigma \mid \sigma \in P_u,\ a \mid_M^\sigma = \mathbf{false}\}$ is also sc(P_u).

The following is an example of such a model:

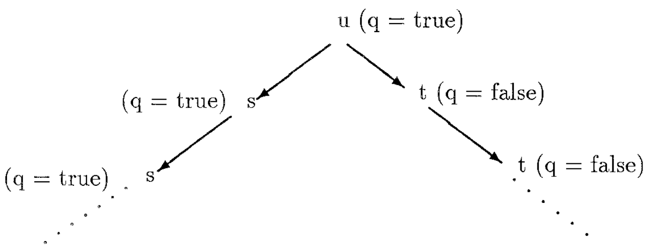

FIG. 1. A model M where $M \not\models \Box q$ and $M \not\models \neg\Box q$.

Example 6.3 (See Fig. 1 above). Suppose $q \in Pvar$. $M = \langle S, u, l, R\rangle$ will be a model where

$$S = \{s, t, u\}$$

$$l(u) = l(s) = Pvar, \qquad l(t) = Pvar - \{q\}$$

$$R = \{(u, s), (u, t), (s, s), t, t)\}.$$

Clearly, $M \not\models \Box q$ and $M \not\models \neg\Box q$.

Definition 6.4. If $a \in \Gamma$, we say that a is *valid* if every model M satisfies a, and we shall denote this fact by $\models a$.

7. THE LOGICAL SYSTEM

The logical system we use is exactly the same system presented by [LS82] for their finite model, with the same numbering. It contains schemata for axioms and rules of inference. An axiom schemata denotes all formulas obtained from it by consistent substitution of arbitrary formulas for the formula variables (a, b, c) appearing in it, and consistent substitution of arbitrary propositional variables for the variables (p, q, ...) that stand for propositional variables. A replacement of a propositional variable by an arbitrary formula is not allowed. The symbol $\vdash$ denotes provability in the system.

The system consists of the following axioms and inference rules. It is not necessarily the most economical.

- The axioms:

(AO) A suitable axiomatization of the propositional calculus.

(A1) $\bigcirc(a \to b) \to \bigcirc a \to \bigcirc b$

(A2) $\neg\bigcirc a \leftrightarrow \bigcirc\neg a$

(A3) $\Box(a \to b) \to \Box a \to \Box b$

(A4) $a\ \mathbf{Until}\ b \to \Diamond b$

(A5) $\Box a \to a \wedge \bigcirc\Box a \wedge \bigcirc a$

(A6) $a\ \mathbf{Until}\ b \leftrightarrow b \vee a \wedge \bigcirc(a\ \mathbf{Until}\ b)$

(A7) $\Box(a \to \bigcirc a) \to a \to \Box a$

(A8) $\nabla(a \to b) \to \nabla a \to \nabla b$

(A9) $\triangle\nabla a \leftrightarrow \nabla a$

(A10) $\nabla a \to a$

(A11) $p \to \nabla p$
(A12) $\nabla \bigcirc a \to \bigcirc \nabla a$
(A14) $\Box \Diamond \triangle a \to \Diamond a$

- The inference rules:

(R0) (Modus Ponens) If $\vdash a$ and $\vdash a \to b$, then $\vdash b$.
(R1) ($\Box$ generalization) If $\vdash a$, then $\vdash \Box a$.
(R2) (∇ generalization) If $\vdash a$, then $\vdash \nabla a$.

Axioms (A0)–(A7) and inference rule (R0) are the logical system presented by [GPSS80] for the temporal logic of linear time, with minor modifications resulting from the fact that their semantics for the connective **Until** is slightly different from ours and is as follows: a **Until** $b \mid_M^{\sigma} = \textbf{true} \Leftrightarrow \exists n \in \mathcal{N}$ such that $b \mid_M^{\sigma^{+n}} = \textbf{true}$ and for all $0 < k < n$, $a \mid_M^{\sigma^{+k}} = \textbf{true}$. Note that in axiom (A11) p stands for a propositional variable and cannot be replaced by an arbitrary formula.

Axiom (A14) intuitively means that if a is possible infinitely many times, a will happen.

8. SOUNDNESS

In this section we will prove the soundness of the logical system given in Section 7. First we shall prove two lemmas which suggest some topological features of the set of paths in which a certain formula holds. We will use the following notations: If a is a formula, $M = \langle S, u, l, R \rangle$ is a model and P is a set of paths in the model, $P(a)$ will denote the subset of paths in P in which a is true, and $P^{+n}(a)$ will denote the set $\{\sigma \mid \sigma \in P,\ a \mid_M^{\sigma^{+n}} = \textbf{true}\}$.

Lemma 8.1. *Let a be a formula, $M = \langle S, u, l, R \rangle$ a model, $n \in \mathcal{N}$. If for all $s \in S$ $P_s(a)$ is almost-open(P_s) then for all $s \in S$ $P_s^{+n}(a)$ is also almost-open(P_s).*

Proof. For all $\tau \in P_s$ there is a set $G_{\tau,n}$ such that $G_{\tau,n}$ is open($P_{\tau(n)}$) and $P_{\tau(n)}(a) \triangle G_{\tau,n}$ is fc($P_{\tau(n)}$) and therefore also fc(P_s). By Lemma 2.5, $\bigcup_{\tau \in P_s} (P_{\tau(n)}(a) \triangle G_{\tau,n}) = (\bigcup_{\tau \in P_s} P_{\tau(n)}(a)) \triangle (\bigcup_{\tau \in P_s} G_{\tau,n})$ is fc(P_s), so $P_s^{+n}(a)$ is almost-open(P_s). ▮

Theorem 8.2. *For every formula a and for every model $M = \langle S, u, l, R \rangle$, $P_u(a)$ is almost-open(P_u).*

Proof. Let $M = \langle S, u, l, R \rangle$ be a model and a a formula. The proof is by induction on the formula structure.

$a = p$: if $p \in l(u)$, then $P_u(a) = P_u$; else $P_u(a) = \emptyset$. In either case, $P_u(a)$ is almost-open.

Suppose that for the formulas b and c, $P_u(b)$ and $P_u(c)$ are almost-open(P_u). We will show that the assertion holds for the following cases:

$a = \neg b$

Since we assume $P_u(b)$ is almost-open, by Lemma 2.11 $P_u(a) = P_u(\neg b)$ is almost-open.

$a = \bigcirc b$

Follows from Lemma 8.1.

$a = \Box b$

Clearly, $P_u(\neg \Box b) = \bigcup_{n \in \mathcal{N}} P_u^{+n}(\neg b)$. Following the case $a = \neg b$ and Lemma 8.1, for every $n \in \mathcal{N}$ $P_u^{+n}(\neg b)$ is almost-open. By Lemma 2.13 $P_u(\neg \Box b)$ is almost-open and therefore (Lemma 2.11) $P_u(\Box b)$ is also almost-open.

$a = \nabla b$

If $P_u(b)$ is co-m(P_u), then $P_u(a) = P_u$; else $P_u(a) = \emptyset$. In either case, $P_u(a)$ is almost-open.

$a = b \vee c$

$P_u(a) = P_u(b) \cup P_u(c)$ According to the induction hypothesis and Lemma 2.13, $P_u(a)$ is almost-open.

$a = b$ **Until** c

$P_u(a) = \bigcup_{n \in \mathcal{N}} \{(\bigcap_{i=0}^{n-1} \text{n-env}(\sigma)^{+i}(b)) \cap \text{n-env}(\sigma)^{+n}(c)\}$. By Lemmas 8.1 and 2.13, $P_u(a)$ is almost-open(P_u). ▮

The meaning of the last theorem was already pointed out in the introduction: If we ignore sets of first category then every property that can be expressed in the language is of finite character.[3]

Theorem 8.3. *For any $a \in \Gamma$, if $\vdash a$ then $\models a$.*

Proof. We will show that if $\vdash a$, then for any model $M = \langle S, u, l, R \rangle$ and state $s \in S$, $P_s(a)$ is co-m(P_s).

Soundness of axioms A0–A11 is easy. The proof can be found in [BE87] and is similar to the soundness proof for these axioms as it appears in [LS82]. The soundness proof for A12 and A14 follows:

(A12) $\nabla \bigcirc a \to \bigcirc \nabla a$

Let $s \in S$, $\tau \in P_s$. Suppose $\bigcirc \nabla a \mid_M^{\tau} = \textbf{false}$. We will show that $\nabla \bigcirc a \mid_M^{\tau} = \textbf{false}$. $\bigcirc \nabla a \mid_M^{\tau} = \textbf{false} \Leftrightarrow \nabla a \mid_M^{\tau^{+1}} = \textbf{false} \Leftrightarrow P_{\tau(1)}(\neg a)$ is sc($P_{\tau(1)}$).

In the natural homeomorphism f between 1-env(τ) and $P_{\tau(1)}$, $f^{-1}(P_{\tau(1)}(\neg a)) = (\text{1-env}(\tau))(\neg \bigcirc a)$, so $(\text{1-env}(\tau))(\neg \bigcirc a)$ is sc(1-env(τ)). Since 1-env(τ) is open(P_s), $(\text{1-env}(\tau))(\neg \bigcirc a)$ is sc(P_s). Since $(\text{1-env}(\tau))(\neg \bigcirc a)$ is a subset of $P_s(\neg \bigcirc a)$, the later is also sc(P_s). So $\forall \sigma \in P_s \nabla \bigcirc a \mid_M^{\sigma} = \textbf{false}$ and especially $\nabla \bigcirc a \mid_M^{\tau} = \textbf{false}$.

(A14) $\Box \Diamond \triangle a \to \Diamond a$

Let $s \in S$. We will show that $A =_{\text{def}} \{\sigma \mid \sigma \in P_s,\ \Box \Diamond \triangle a \mid_M^{\sigma} = \textbf{true}$, and $\Diamond a \mid_M^{\sigma} = \textbf{false}\}$ is fc(P_s). We will use Lemma 2.6. Let G be open(P_s). We will show that there exists an open set G', $G' \subseteq G$, such that $G' \cap A$ is fc(P_s).

[3] We remind the reader that a property has a finite character iff whenever a general sequence has that property then there is a finite initial segment of it, such that every general sequence extending this initial segment has the property.

If $G \cap A = \varnothing$, then we will choose $G' = G$; else there exists $\tau \in G \cap A$. Since G is open, there exists $n \in \mathcal{N}$ such that the n-environment of τ is in G.

Since $\Box \Diamond \triangle a \,|_M^{\tau} = \mathbf{true}$, there exists $m \geqslant n$ such that $\triangle a \,|_M^{\tau^{+m}} = \mathbf{true}$. That means that $P_{\tau(m)}(a)$ is $\mathrm{sc}(P_{\tau(m)})$. According to Theorem 8.2, there exists a set E which is $\mathrm{open}(P_{\tau(m)})$ and sets C, D which are $\mathrm{fc}(P_{\tau(m)})$ such that $P_{\tau(m)}(a) = (E \cup C) - D$. Since $P_{\tau(m)}(a)$ is $\mathrm{sc}(P_{\tau(m)})$, E is not empty.

Assertion A. *$P_{\tau(m)}(\neg a) \cap E$ is* $\mathrm{fc}(P_{\tau(m)})$.

Proof. Suppose conversely that $P_{\tau(m)}(\neg a) \cap E$ is $\mathrm{sc}(P_{\tau(m)})$; so for every set $C \subseteq P_{\tau(m)} P_{\tau(m)}(\neg a) \cap (E \cup C)$ is $\mathrm{sc}(P_{\tau(m)})$, and so for every set D which is $\mathrm{fc}(P_{\tau(m)})$, $[(E \cup C) - D] \cap P_{\tau(m)}(\neg a)$ is $\mathrm{sc}(P_{\tau(m)})$ and therefore not empty. So we cannot find C and D which are $\mathrm{fc}(P_{\tau(m)})$ and $P_{\tau(m)}(a) = (E \cup C) - D$, a contradiction. ▌ (Assertion A)

The natural homeomorphism f between n-env(τ) and $P_{\tau(m)}$ implies that $f^{-1}(P_{\tau(m)}(\neg a)) = (m\text{-env}(\tau)^{+m})(\neg a)$, $(m\text{-env}(\tau)^{+m})(\neg a) \cap f^{-1}(E)$ is $\mathrm{fc}(n\text{-env}(\tau))$ and therefore $\mathrm{fc}(P_s)$, and $f^{-1}(E)$ is $\mathrm{open}(n\text{-env}(\tau))$ and not empty. Clearly, $A \cap n\text{-env}(\tau) \subseteq (m\text{-env}(\tau)^{+m})(\neg a)$, and therefore $A \cap n\text{-env}(\tau) \cap f^{-1}(E) = A \cap f^{-1}(E)$ is $\mathrm{fc}(P_s)$. Since $n\text{-env}(\tau) \subseteq G$ and n-env(τ) is $\mathrm{open}(P_s)$, $f^{-1}(E)$ is an open subset of G. We will choose $G' = f^{-1}(E)$.

Soundness of the inference rules:

(R0) (Modus Ponens)—obvious.

(R1) If $\vdash a$, then $\vdash \Box a$

Suppose that for every model M, $M \models a$. We want to show that for every model M, $M \models \Box a$. Let $M = \langle S, u, l, R \rangle$ be an arbitrary model. Clearly, $P_u(\neg \Box a) = \bigcup_{n \in \mathcal{N}} P_u^{+n}(\neg a)$. We will show that for every n, $P_u^{+n}(\neg a)$ is $\mathrm{fc}(P_u)$. Let $n \in \mathcal{N}$, $\tau \in P_u$. In the natural homeomorphism f between n-env(τ) and $P_\tau(n)$, $f^{-1}(P_{\tau(n)}(\neg a)) = n\text{-env}(\tau)^{+n}(\neg a)$. Since $\models a$ $P_{\tau(n)}(\neg a)$ is $\mathrm{fc}(P_\tau(n))$ and therefore $n\text{-env}(\tau)^{+n}(\neg a)$ is $\mathrm{fc}(n\text{-env}(\tau))$. Since n-env(τ) is $\mathrm{open}(P_u)$, $n\text{-env}(\tau)^{+n}(\neg a)$ is $\mathrm{fc}(P_u)$.

Let $E \subseteq P_u$ be the maximal set such that for every $\tau, \sigma \in E$ $\sigma \neq \tau \Rightarrow n\text{-env}(\tau) \cap n\text{-env}(\sigma) = \varnothing$. By Lemma 2.5. $\bigcup_{\sigma \in E} n\text{-env}(\tau)^{+n}(\neg a)$ is $\mathrm{fc}(P_u)$. Clearly, $P_u^{+n}(\neg a) = \bigcup_{\sigma \in E} n\text{-env}(\tau)^{+n}(\neg a)$.

(R2) If $\vdash a$, then $\vdash \nabla a$

R2 is obviously sound in light of Lemma 6.2. ▌

9. COMPLETENESS

Our completeness proof is a variation of the completeness proof presented by [LS82].

The basic idea of the proof goes back to the proof of the completeness of temporal logic by [GPSS80]. The idea is to try to use states which are partial theories expressing what the sequences starting from this state should satisfy. In order to satisfy a ∇b formula at a path σ which requires b to be false, we have to construct many sequences starting from the first state of σ in which b is true (this motivates the "alternative" relation).

In the first subsection we repeat results from [LS82].

9.1. Results from Lemann and Shelah's Paper

We will use the following theorem:

(T3) $\vdash \Diamond a \leftrightarrow a \vee \bigcirc \Diamond a$

9.1.1. *Theories, Traces, and Relations among Them*

9.1.1.1. *Theories*

Definition 9.1. A *theory* is any subset of Γ.

Definition 9.2. A theory T is said to be *inconsistent* if there is $n \in \mathcal{N}$ and formulas $a_0, a_1, ..., a_n \in T$ such that $\vdash a_1 \wedge \ldots \wedge a_n \rightarrow \neg a_0$. If T is not inconsistent, it is said to be *consistent.*

Definition 9.3. A theory T is said to be *complete* if for any formula $a \in \Gamma$, either $a \in T$ or $\neg a \in T$.

Lemma 9.4. *If T is a consistent and complete theory, then*

(a) *if $a \in T$ and $\vdash a \rightarrow b$, then $b \in T$;*

(b) *$a \in T \Leftrightarrow \neg a \notin T$;*

(c) *$a \vee b \in T \Leftrightarrow a \in T$ or $b \in T$;*

(d) *if T' is a consistent theory (not necessarily complete), then there is a consistent and complete theory T such that $T' \subseteq T$;*

(e) *$\nvdash a \Leftrightarrow$ there is a consistent and complete theory T such that $\neg a \in T$.*

9.1.1.2. *Relations among Theories*

The successor relation

Definition 9.5. Let T_1 and T_2 be two theories. We say that T_2 is a *successor* of T_1 and write $T_1 \rho T_2$ if $\forall a \in \Gamma$ such that $\bigcirc a \in T_1$, we have $a \in T_2$.

Lemma 9.6. *If T is consistent and complete theory, there is a unique consistent and complete theory T^+ such that $T \rho T^+$. It is characterized by $a \in T^+ \Leftrightarrow \bigcirc a \in T$.*

The future relation

Definition 9.7. We say that T_2 is a *future* of T_1 and write $T_1 \leqslant T_2$ if $\forall a \in \Gamma$ such that $\Box a \in T_1$, we have $a \in T_2$.

Lemma 9.8. *Among consistent and complete theories the relation $\leqslant$ is reflexive and transitive; it contains the relation ρ.*

Lemma 9.9. *Let T be a consistent and complete theory and $a \in \Gamma$ a formula with $\neg \Box a \in T$. There is a consistent and complete theory T', such that $T \leqslant T'$ and $\neg a \in T'$.*

The alternative relation

Definition 9.10. We say that T_2 is an *alternative* for T_1 and write $T_1 \equiv T_2$ if $\forall a \in \Gamma$ such that $\nabla a \in T_1$, we have $a \in T_2$.

Lemma 9.11. *Among consistent and complete theories the relation $\equiv$ is an equivalence relation.*

A consequence is that, if $T \equiv T'$, then $\nabla a \in T$ iff $\nabla a \in T'$.

Lemma 9.12. *Let T be a consistent and complete theory and $a \in \Gamma$ a formula with $\neg \nabla a \in T$. There is a consistent and complete theory T' such that $T \equiv T'$ and $\neg a \in T'$.*

Whenever R_1 and R_2 are relations, $R_1 R_2$ will denote the composition of the two relations (R_1 first, and then R_2).

Lemma 9.13. *Let $k \geqslant 0$ and $T_0, T_1, ..., T_k$ be consistent and complete theories such that $\forall i$, $0 \leqslant i < k$, $T_i \equiv \rho T_{i+1}$. There are consistent and complete theories V_i, for $i = 0, ..., k$, such that*

1. $V_k = T_k$,
2. $V_i \equiv T_i$, $\forall i$, $0 \leqslant i \leqslant k$,
3. $V_i \rho V_{i+1}$, $\forall i$, $0 \leqslant i < k$.

9.1.1.3. *Therminal theories and Terminal Relations*

Terminal theories

Definition 9.14. A consistent and complete theory T is said to be *terminal* iff it satisfies any one of the two equivalent properties:

1. $\forall a$, $a \in \Gamma$, $\Diamond \Box a \in T \Rightarrow a \in T$,
2. $\forall a$, $a \in \Gamma$, $\Diamond a \in T \Rightarrow \Box \Diamond a \in T$.

Lemma 9.15. 1. *Let T and T' be consistent and complete theories, such that $T \leqslant T'$, then, if T is terminal, so is T'.*

2. *Let T be a consistent, complete, and terminal theory, then $T^+ \leqslant T$.*

Lemma 9.16. *Let T be a consistent and complete theory, then there is a consistent and complete terminal theory T', such that $T \leqslant T'$.*

Terminal relations

Definition 9.17. Let T_1 and T_2 be consistent and complete. We say that T_2 is a terminal alternative for T_1 and write $T_1 \approx T_2$ iff

1. $T_1 \equiv T_2$,
2. $T_1 \leqslant T_2$,
3. $T_2 \leqslant T_1$.

Conditions (2) and (3) together are equivalent to : $\forall a \in \Gamma$, $\Box a \in T_1 \Leftrightarrow \Box a \in T_2$.

Note that by Lemma 9.15, if T_1 is terminal and $T_1 \approx T_2$, then T_2 is also terminal.

Lemma 9.18. *Among consistent and complete theories, the relation $\approx$ is an equivalence relation: it is contained in the relation $\equiv$.*

9.1.1.4. *Basic Lemmas*

Lemma 9.19. *Let T be a consistent complete terminal theory and $a \in \Gamma$ a formula with $\neg \nabla a \in T$. There is a consistent and complete theory T', such that $T \approx T'$ and $\neg a \in T'$. The theory T' is terminal.*

Lemma 9.20. *Let $k \geqslant 0$ and $T_0, T_1, ..., T_k$ be consistent and complete terminal theories such that $\forall i$, $0 \leqslant i < k$, $T_i \approx \rho T_{i+1}$. There are consistent and complete terminal theories V_i, $i = 0, ..., k$ such that*

1. $T_k = V_k$
2. $V_i \approx T_i$, $\forall i$, $0 \leqslant i \leqslant k$
3. $V_i \rho V_{i+1}$, $\forall i$, $0 \leqslant i < k$.

9.1.1.5. *Traces and Relations among Them*

Since not every consistent theory has a model, we need to restrict our attention to a finite set of formulas. Withour loss of generality, from now on we will assume that *Pvar* is finite set. Therefore, for every $n \in \mathcal{N}$, Γ_n is finite, where we define Γ_n to be the set of all formulas of size less or equal to n. We are interested only in the formulas of Γ_n, but for the completeness proof we need to consider also some larger formulas. So we define n' to be slightly larger than n, for example, we define $n' =_{\text{def}} 3n + 10$. We define traces of theories over $\Gamma_{n'}$, but we'll claim only about formulas of Γ_n.

Definition 9.21. A trace D of size n is the intersection of some consistent and complete theory T with $\Gamma_{n'}$.

Note that we use here n' not n. Since in our proof n will be fixed, we will usually use simply the term "trace" instead of "trace of size n." Let $\mathcal{D}_n$ be the set of all traces of size n. $\mathcal{D}_n$ is a finite set.

We define between traces all the relations that have been defined between theories, $\rho, \leqslant, \equiv, \equiv \rho, \approx \rho$, and $\approx$, in the following way: $DRD' \Leftrightarrow$ there are theories T and T' such that $D = T \cap \Gamma_{n'}$, $D' = T' \cap \Gamma_{n'}$, and TRT'. Note that $\equiv \rho$ and $\approx \rho$ are the projections of the composition of relations, not the composition of the projections. Similarly, terminal traces are traces of terminal theories.

We define ρ^* to be the reflexive and transitive closure of the relation ρ on traces. Note that ρ^* is the reflexive and transitive closure of the projection, not the projection of the reflexive and transitive closure.

Lemma 9.22. *Let D and D' be traces. If $D \leqslant D'$, then $D\rho^* D'$.*

9.2. New Results

Some more theorems about traces:

Lemma 9.23. *Let D be a trace, and let $a \in \Gamma$ be a formula such that $\Diamond a \in D$. There exists a trace D' such that $D\rho^* D'$ and $a \in D'$.*

Proof. Suppose $\neg \Box \neg a \in D$. By Lemma 9.9, there exists a trace D' such that $D \leqslant D'$ and $a \in D'$. By Lemma 9.22 $D\rho^* D'$. ▮

Lemma 9.24. *Let $D_0, D_1, ..., D_k$ be traces such that $\forall i$, $0 \leqslant i < k$, $D_i \rho D_{i+1}$ and $a \in \Gamma$ is a formula such that $\Diamond a \in D_0$ and $\#(\Diamond a) < n'$. If $\forall i$, $0 \leqslant i < k$, $a \notin D_i$, then $\Diamond a \in D_k$.*

Proof. By induction on k:

Case $k = 0$. Obvious.

Case $k > 0$. By the induction hypothesis $\Diamond a \in D_{k-1}$. By (T3), and since $a \notin D_{k-1}$ and $\#(\Diamond a) < n'$, $\bigcirc \Diamond a \in D_{k-1}$. Since $D_{k-1} \rho D_k$, $\Diamond a \in D_k$. ▮

Lemma 9.25. *Let $m \in \mathcal{N}$, $m < n'$. Let D_0 be a trace. There exist traces $D_1, ..., D_k$ such that $\forall i$, $0 \leqslant i < k$ $D_i \rho D_{i+1}$ and $\forall a \in \Gamma$ such that $\#(\Diamond a) \leqslant m$ and $\Diamond a \in D_0$ $\exists j < k$ such that $a \in D_j$.*

Proof. Let $m < n'$, $A =_{\text{def}} \{a \mid \Diamond a \in D_0,\ \#(\Diamond a) \leqslant m\}$. Since D_0 is a finite set of formulas, $\exists n \in \mathcal{N}$ such that $|A| = n$. So we can arrange all the members of A in an order $a_0, ..., a_n$.

The proof goes by induction on n:

Case $n = 0$. Since $\Diamond a_0 \in D_0$, by Lemma 9.23 there exist a trace D' such that $D_0 \rho^* D'$ and $a \in D'$. By Lemma 9.6 there exists a trace E such that $D' \rho E$.

Case $n > 0$. By the induction hypothesis, there exist $j \in \mathcal{N}$ and traces $D_1, ..., D_j$ such that $\forall i$, $0 \leqslant i < j$ $D_i \rho D_{i+1}$ and for every a_i, $0 \leqslant i \leqslant n-1$ $\exists l_i < j$ such that $a_i \in D_{l_i}$. If $\exists l < j$ such that $a_n \in D_l$, we are done; else by Lemma 9.24, $\Diamond a_n \in D_j$. By Lemma 9.23, there exists a trace D' such that $D_j \rho^* D'$ and $a_n \in D'$. By Lemma 9.6, there exists a trace E such that $D' \rho E$. ▮

9.3. The Completeness Proof

Theorem 9.26. *For any $a \in \Gamma$, if $\models a$ then $\vdash a$.*

Proof. Suppose that $\not\vdash a$; we will build a model that does not satisfy a. By Lemma 9.4(e), there is a consistent and complete theory T_a, that contains $\neg a$. We define $n = \#(a)$ and look at traces of size n. Let $D_a =_{\text{def}} T_a \cap \Gamma_{n'}$. D_a is a trace of size n that contains $\neg a$.

The model $M = \langle S, u, l, R \rangle$ that does not satisfy a is defined in the following way:

- $S = \mathscr{D}_n$
- $u = D_a$
- $l(D) = \{p \mid p \in D\}$
- Let $D \in S$. If D is not terminal, then $\forall E \in S$, $R(D, E)$ iff $D \equiv \rho E$. If D is terminal, then $\forall E \in D$, $R(D, E)$ iff $D \approx \rho E$.

Note that since $\equiv$ and $\approx$ are reflexive, $\forall E, D \in S$ $E \rho D \Rightarrow R(E, D)$, and since (by Lemma 9.6) for each trace D there is a trace E such that $D \rho E$, the model satisfies: $\forall D \in S$ $\exists E$ such that $R(D, E)$.

Note that if D is terminal and $R(D, E)$ then E is also terminal, according to Definition 9.17 and Lemmas 9.8 and 9.15.

If $D \rho E$ we will call the transition from D to E a ρ-transition.

From now on, we will continue using the notation s, t, u, ... for the states of the model, keeping in mind that each state is actually a trace of size n.

Definition 9.27. Let $m, n \in \mathcal{N}$. Let σ be a path in M. We shall say that σ is an m, n-*standard* path if there exist indices $i_0 < i_1 \cdots < i_m$ such that the following condition hold:

1. $i_0 = 0$.
2. $\sigma(i_1)$ is terminal.
3. $\forall j$ such that $1 \leqslant j \leqslant m$ and $\forall a \in \Gamma$ such that $\#(\Diamond a) \leqslant n$ and $\Diamond a \in \sigma(i_{j-1})$, exist $i_{j-1} \leqslant k < i_j$ such that $a \in \sigma(k)$.
4. $\forall j$, $0 \leqslant j < i_m$, $\sigma(j) \rho \sigma(j+1)$

Our definition of an m-standard path modifies the one given by [LS82]:

Definition 9.28. Let σ be a path in M. We will say that σ is m-*standard* if it is m, m-standard.

Note that:

1. If a path σ is m-standard, then $\forall i \geqslant i_1$ (i_1 as in Definition 9.28 above) $\sigma(i)$ is terminal.
2. If a path σ is m-standard, then $\forall i \in \mathcal{N}$ $\sigma(i) \approx \rho \sigma(i+1)$.

Lemma 9.29. *Let $m < n'$, and let $i_0, i_1, ..., i_m$ witness the fact that σ is m-standard according to Definition* 9.28. *For all i, $0 \leqslant i \leqslant i_1$, and $\forall k < m$, σ^{+i} is k-standard.*

Proof. Let $0 \leqslant i \leqslant i_1$. It is enough to show that σ^{+i} is $(m-1)$-standard. We claim that $i, i_2, i_3, ..., i_m$ witness the fact that σ^{+i} is $(m-1)$-standard. Conditions 1, 2, and 4 of Definition 9.28 clearly hold for these indices. It is left to prove that condition 3 holds for them. Clearly, $\forall l$ such that $2<l\leqslant m$ and $\forall a\in\Gamma$ such that $\#(\Diamond a)\leqslant m$ and $\Diamond a\in\sigma(i_{l-1})$, exist $i_{l-1}\leqslant k<i_l$ such that $a\in\sigma(k)$.

Now, suppose $\#(\Diamond a)\leqslant m-1$ and $\Diamond a\in\sigma(i)$. If $i=i_1$, then since σ is m-standard there must exist $i\leqslant l<i_2$ such that $a\in\sigma(l)$. If $i<i_1$ and there is no such l which is smaller than i_1, it must be (Lemma 9.24) that $\Diamond a\in\sigma(i_1)$, and since σ is m-standard there must be such l.

Definition 9.30. A path σ is said to be *ultimately m-standard* if there is $n\in\mathcal{N}$ such that σ^{+n} is m-standard.

Lemma 9.31. *Let $m\in\mathcal{N}$, $m<n'$, $s\in S$. For every $0\leqslant j\leqslant m$ there exists a path $\tau_j\in P_s$ such that τ_j is j, m-standard.*

Proof. Let $m\in\mathcal{N}$. The proof goes by induction on j.

Case $j=0$. Obvious.

Case $j=1$. We will define $\tau_0(0)=s$. By Lemma 9.25 there exist traces $D_1, ..., D_k$ such that $\forall n$, $1\leqslant n<k$ $D_n\rho D_{n+1}$, $s\rho D_1$, and $\forall a\in\Gamma$ such that $\#(\Diamond a)\leqslant m$ and $\Diamond a\in s$ $\exists n<k$ such that $a\in D_n$. We will define $\forall n$, $1\leqslant n\leqslant k$, $\tau_0(n)=D_n$. If D_k is terminal, we will choose $i_1=k$; else, by Lemma 9.16, $\exists t$ such that t is terminal and $D_k\leqslant t$. By Lemma 9.22 there exist traces $E_0, ..., E_l$ such that $\forall i$, $0\leqslant i<l$, $E_i\rho E_{i+1}$, $E_0=D_k$, and $E_l=t$. We will define $\forall i$ $0\leqslant i\leqslant l$, $\tau_0(k+i)=E_i$, and choose $i_1=k+l$.

Case $j>1$. By the induction hypothesis, there exists a path $\tau_{j-1}\in P_s$ such that τ_{j-1} is $(j-1)$, m-standard. By Lemma 9.25 there exist traces $D_1, ..., D_n$ such that $\forall i$, $1\leqslant i<n$, $D_i\rho D_{i+1}$, $\sigma(i_{j-1})\,\rho D_1$, and $\forall a\in\Gamma$ such that $\#(\Diamond a)\leqslant m$ and $\Diamond a\in\sigma(i_{j-1})$ $\exists i<n$ such that $a\in D_i$. We will define $\forall i$, $0\leqslant i\leqslant i_{j-1}$, $\tau_j(i)=\tau_{j-1}(i)$; $\forall i$, $1\leqslant i\leqslant n$, $\tau_j(i_{j-1}+i)=D_i$. ∎

Lemma 9.32. *Let $s\in S$, $m\in N$, $1\leqslant m<n'$. If A is the set of m-standard paths beginning at s, then* Int $A\neq\varnothing$.

Proof. By Lemma 9.31 above $\exists\tau$ such that τ is m-standard. The i_m-environment of τ is open(P_s) and it is a subset of A. ∎

Lemma 9.33. *Let $s\in S$, $m\in\mathcal{N}$, $m<n'$. The set $\{\sigma\mid\sigma\in P_s$, σ is ultimately m-standard$\}$ is co-m(P_s).*

Proof. We will show that the set $\{\sigma\mid\sigma\in P_s, \sigma$ is *not* ultimately m-standard$\}$ is nowhere-dense, by proving that for every set $G\neq\varnothing$ which is open(P_s) there is a set G' such that $G'\subseteq G$, $G'\neq\varnothing$, G' is open(P_s), and $\forall\sigma\in G'$, σ is ultimately m-standard.

Let G be open(P_s), and suppose that $G\neq\varnothing$. Let $\tau\in G$. Since G is open, $\exists n$ such that the n-environment of τ is in G. By Lemma 9.32, there is a set $D\neq\varnothing$ which is open$(P_{\tau(n)})$ and for every $\sigma\in D$ σ is m-standard.

Define $G'=_{\text{def}}\{\sigma\mid\sigma\in n\text{-env}(\tau),\quad\exists\sigma'\in D\quad\sigma'=\sigma^{+n}\}$. Clearly, $G'\subseteq G$, $G'\neq\varnothing$, and each $\sigma\in G'$ is ultimately m-standard. By Lemma 3.7, G' is open. ∎

The definitions of *generic path* and *equivalent paths* are the same as in [LS82]:

Definition 9.34. Let σ and τ be paths in M. σ and τ are said to be *equivalent* (notation: $\sigma\equiv\tau$) if $\forall i\in\mathcal{N}$, $\sigma(i)\equiv\tau(i)$.

Definition 9.35. Let σ be a path in M. σ is said to be generic if for any trace s_0 that appears an infinite number of times in σ and for any finite sequence of traces $s_0, s_1, ..., s_m$ such that $s_i\rho s_{i+1}$ for every i such that $0\leqslant i<m$, the sequence above appears in σ an infinite number of times.

Note that if a path is generic, and if a trace s appears an infinite number of times in the path, then every trace t such that $s\rho^*t$ also appears an infinite number of times in the path.

The following lemma corresponds to [LS82] claim that a generic sequence has probability one.

Lemma 9.36. *Let $s\in S$. The set $\{\sigma\mid\sigma\in P_s$, σ is generic$\}$ is co-m(P_s).*

Proof. Let $s\in S$. We will show that the set $\{\sigma\mid\sigma\in P_s, \sigma$ is *not* generic$\}$ is fc(P_s).

$\{\sigma|\sigma\in P_s,\ \sigma$ is *not* generic$\}=\bigcup_{t\in S}\{\bigcup_{m\in\mathcal{N}}\{\bigcup_{x\in B_{t,m}}\{\bigcup_{n\in\mathcal{N}}A_{t,x,n}\}\}\}$, where $B_{t,m}=_{\text{def}}\{x\mid x$ is a sequence of $m+1$ traces $s_0, ..., s_m$ such that $s_0=t$ and $\forall i$, $0\leqslant i<m$, $s_i\rho s_{i+1}\}$, and $A_{t,x,n}=_{\text{def}}\{\sigma\mid\sigma\in P_s$, t appears in σ an infinite number of times, but x appears exactly n times in $\sigma\}$. Note that $S=\mathscr{D}_n$ is a finite set, so it is enough to show that for every $t\in S$, $m\in N$, $n\in\mathcal{N}$, and $x\in B_{t,m}$, $A_{t,x,n}$ is nowhere dense.

Let G be open(P_s), and suppose $G\neq\varnothing$ and $G\cap A_{t,x,n}\neq\varnothing$. Let $\tau\in G\cap A_{t,x,n}$. $\exists n_1$ such that the sequence x does not appear after $\tau(n_1)$. $\exists n_2$ such that the n_2-environment of τ is in G. Since t appears an infinite number of times in τ, $\exists k$, $k>\text{MAX}(n_1,n_2)$ such that $\sigma(k)=t$. Since $A_{t,x,n}$ is not empty, we have traces $s_0, ..., s_m$, which form the sequence x, such that $\forall i$, $0\leqslant i<m$, $s_i\rho s_{i+1}$.

We will define the following path τ' (we define only its beginning, and as in previous proofs, this is enough):

- $\forall i$, $0\leqslant i\leqslant k$ $\tau'(i)=\tau(i)$,
- $\forall i$, $0\leqslant i\leqslant m$ $\tau'(k+i)=s_i$.

There is a path τ' that begins in this way. Since τ' is in the n_2-environment of τ, $\tau'\in G$.

Since G is open, $\exists j$, $j>k+m$ such that $C=_{\text{def}}\{$the j-environment of $\tau'\}$ is in G. C is open, not empty, and in every path in C the sequence x appears at least $n+1$ times. So $C\cap A_{t,x,n}=\varnothing$. ∎

Lemma 9.37. *Let $s\in S$, $m\in\mathcal{N}$, $m<n'$. The set of generic m-standard paths beginning at s is* sc(P_s).

Proof. Let $A=_{\text{def}}\{\sigma\mid\sigma\in P_s,\quad\sigma$ is generic$\}$, $B=_{\text{def}}\{\sigma\mid\sigma\in P_s,\ \sigma$ is m-standard$\}$. We have proved that A is co-m(P_s) and B is sc(P_s), so by Lemma 2.4 $A \cap B$ is sc(P_s). ▮

Lemma 9.38. *Let $b \in \Gamma_n$, σ a $\#(\Diamond b)$-standard generic path of M and τ and τ' be two equivalent paths of M; then*

(a) $b \mid^{\sigma}_{M} = \mathbf{true} \Leftrightarrow b \in \sigma(0)$

(b) $b \mid^{\tau}_{M} = \mathbf{true} \Leftrightarrow b \mid^{\tau'}_{M} = \mathbf{true}$.

Proof. The proof goes by induction on the size of b (i.e., $\#(b)$). At each induction step we will prove (a) and then (b). We define $m = \#(\Diamond b)$ and throughout the proof we let $i_0, i_1, ..., i_m$ witness the fact that σ is m-standard according to Definition 9.28. The cases $b = p$, $b = \neg c$, and $b = c \vee d$ are done in a way similar to [LS82]. We will elaborate on the other cases:

$b = \bigcirc c$. (a) Suppose $\bigcirc c \in \sigma(0)$. Since $m \geqslant 1$, by Definition 9.27, $\sigma(0)\, \rho \sigma(1)$, so $c \in \sigma(1)$. Since σ is m-standard and $\#(\Diamond c) < m$, σ^{+1} is $\#(\Diamond c)$-standard (Lemma 9.29). Also, σ^{+1} is generic since σ is, so by the induction hypothesis $c \mid^{\sigma^{+1}}_{M} = \mathbf{true}$, so $\bigcirc c \mid^{\sigma}_{M} = \mathbf{true}$.

Suppose $\bigcirc c \notin \sigma(0)$. For the same reasons as before, $c \mid^{\sigma^{+1}}_{M} = \mathbf{false} \Rightarrow \bigcirc c \mid^{\sigma}_{M} = \mathbf{false}$.

(b) $\bigcirc c \mid^{\tau}_{M} = \mathbf{true} \Leftrightarrow c \mid^{\tau^{+1}}_{M} = \mathbf{true} \Leftrightarrow$ (Since $\tau \equiv \tau'$, $\tau^{+1} \equiv \tau'^{+1}$, so we can use the induction hypothesis) $c \mid^{\tau'^{+1}}_{M} = \mathbf{true} \Leftrightarrow \bigcirc c \mid^{\tau'}_{M} = \mathbf{true}$.

$b = \Box c$. (a) 1. Suppose $\Box c \in \sigma(0)$. We will show that $\Box c \mid^{\sigma}_{M} = \mathbf{true}$.

Claim. $\forall i$, $\Box c \in \sigma(i)$.

Proof. This will be shown by induction on i. For $i = 0$, $\Box c \in \sigma(i)$ by the assumption. Now, suppose $\Box c \in \sigma(i)$. By Definition 9.28, $\sigma(i) \approx \rho \sigma(i+1)$; i.e., there exists a trace E such that $\sigma(i) \approx E$ and $E \rho \sigma(i+1)$, so $\Box c \in \sigma(i) \Rightarrow \Box c \in E$. Using (A5), $\Box c \rightarrow \bigcirc \Box c$, and since $n' \geqslant n+1$, $\bigcirc \Box c \in E$. Since $E \rho \sigma(i+1)$ we get $\Box c \in \sigma(i+1)$. ▮

As a corollary to the above claim, $\forall i \in \mathcal{N}$, $c \in \sigma(i)$, because by (A5) $\Box c \rightarrow c$. We will show that $\forall i \in \mathcal{N} c \mid^{\sigma^{+i}}_{M} = \mathbf{true}$. Let $i \in \mathcal{N}$. We will distinguis between two cases:

Case $\sigma(i)$ is not terminal. We will show that in this case the path σ^{+i} is generic and $\#(\Diamond c)$-standard, and since $c \in \sigma^{+i}(0)$, by the induction hypothesis $c \mid^{\sigma^{+i}}_{M} = \mathbf{true}$. We know that σ is a generic m-standard path, and since $\sigma(i)$ is not terminal, it must be that $i < i_1$ (see Definition 9.28), so since $\#(c) < \#(\Box c)$, σ^{+i} is $\#(\Diamond c)$-standard. Since σ is generic, σ^{+i} is also generic.

Case $\sigma(i)$ is terminal. In this case we will show a generic m-standard path τ such that $\tau \equiv \sigma^{+i}$. We will use the induction hyposthesis, part (a), to show that $c \mid^{\tau}_{M} = \mathbf{true}$, and then, since $\tau \equiv \sigma^{+i}$, it will follow from the induction hypothesis, part (b), that $c \mid^{\tau}_{M} = c \mid^{\sigma^{+i}}_{M} = \mathbf{true}$.

Since S is finite, $\exists s \in S$ such that s appears an infinite number of times in σ. Following Lemmas 9.25 and 9.6 there is a sequence of states $s_0, s_1, ..., s_k$ such that $s_0 = s$ and for all $0 \leqslant l < k$, $s_i \rho s_{i+1}$, and there are indices $i_0 = 0, i_1, ..., i_m \in \{1, ..., k\}$ for which condition 3 in Definition 9.28 holds. Since σ is generic $\exists j > i$ such that $\sigma^{+(i+j)}$ begins with this sequence. Clearly, $\sigma^{+(i+j)}$ is generic and $\#(\Diamond c)$-standard. Since $\sigma(i)$ is terminal, $\forall 0 \leqslant k \leqslant j$, $\sigma(i+k)$ is also terminal. By Lemma 9.20, then, we can find traces E_k for $0 \leqslant k \leqslant j$ such that:

1. $E_j = \sigma(i+j)$
2. $E_k \approx \sigma(i+k)$, $\forall k$, $0 \leqslant k \leqslant j$
3. $E_k \rho E_{k+1}$, $\forall k$, $0 \leqslant k < j$.

Let the path τ be defined by

- $\forall k$, $0 \leqslant k \leqslant j$, $\tau(k) = E_k$.
- $\forall k$, $k > j$, $\tau(k) = \sigma(i+k)$.

τ is generic and $\#(\Diamond c)$-standard. So by the induction hypothesis $c \mid^{\tau}_{M} = \mathbf{true} \Leftrightarrow c \in \tau(0)$. But $\tau(0) = E_0$ and $E_0 \approx \sigma(i)$. Since $\Box c \in \sigma(i)$, $\Box c \in \tau(0)$, so also $c \in \tau(0)$, so $c \mid^{\tau}_{M} = \mathbf{true}$. Since $\tau \equiv \sigma^{+i}$, by the induction hypothesis part (b) $c \mid^{\sigma^{+i}}_{M} = \mathbf{true}$.

2. Suppose $\Box c \notin \sigma(0)$. We will show that, $\Box c \mid^{\sigma}_{M} = \mathbf{false}$.

Since $\Box c \notin \sigma(0)$ and $n' > n+3$, $\Box \neg c \in \sigma(0)$. Since $\#(\neg c) = \#(\Box c)$ and σ is $\#(\Diamond \Box c)$-standard, $\exists i < i_1$ such that $\neg c \in \sigma(i) \Rightarrow c \notin \sigma(i)$. Since $i < i_1$ and $\#(c) < \#(\Box c)$, σ^{+i} is $\#(\Diamond c)$-standard. Obviously, σ^{+i} is generic, so by the induction hypothesis, $c \mid^{\sigma^{+i}}_{M} = \mathbf{false} \Rightarrow \Box c \mid^{\sigma}_{M} = \mathbf{false}$.

(b) $\Box c \mid^{\tau}_{M} = \mathbf{true} \Leftrightarrow \forall i \in \mathcal{N}$, $c \mid^{\tau^{+i}}_{M} = \mathbf{true} \Leftrightarrow$ (since $\tau \equiv \tau'$, $\forall i$, $\tau^{+i} \equiv \tau'^{+i}$, so we can use the induction hypothesis part (b)) $\forall i \in \mathcal{N}$ $c \mid^{\tau'^{+1}}_{M} = \mathbf{true} \Leftrightarrow \Box c \mid^{\tau'}_{+} = \mathbf{true}$.

$b = c$ **Until** d. (a) Suppose c **Until** $d \in \sigma(0)$. Then, by (A4), $\Diamond d \in \sigma(0)$. Since $\#(\Diamond d) < \#(\Diamond c$ **Until** $d)$ and σ is $\#(\Box c$ **Until** $d)$-standard, $\exists k < i_1$ such that $d \in \sigma(k)$. Let i be the smallest such k.

Claim. *$\forall j$, $0 \leqslant j < i$ $c \in \sigma(j)$ and $\bigcirc(c$ **Until** $d) \in \sigma(j)$.*

Proof. By induction on j:

Case $j = 0$. By (A6), c **Until** $d \in \sigma(0) \Rightarrow (n' \geqslant 2n)$ $d \vee (c \wedge \bigcirc(c$ **Until** $d)) \in \sigma(0) \Rightarrow (d \notin \sigma(j))$ $c \wedge \bigcirc(c$ **Until** $d) \in \sigma(0)$.

Case $j > 0$. Since $j - 1 < i_1$, $\sigma(j-1)\, \rho \sigma(j)$. By the induction assumption $\bigcirc(c$ **Until** $d) \in \sigma(j-1) \Rightarrow c$ **Until** $d \in \sigma(j)$, and we conclude as in the case $j = 0$.

Now, since $\#(c) < \#(c$ **Until** $d)$, $\#(d) < \#(c$ **Until** $d)$ and $i < i_1$, we can use the induction hypothesis to show that $\forall j$, $0 \leqslant j < i$, $c \mid^{\sigma^{+j}}_{M} = \mathbf{true}$ and $d \mid^{\sigma^{+i}}_{M} = \mathbf{true} \Rightarrow c$ **Until** $d \mid^{\sigma}_{M} = \mathbf{true}$.

Suppose c **Until** $d \notin \sigma(0)$. Then, by (T7), and since $n' > 3n$, $\Box\neg d \vee [\neg d$ **Until** $(\neg c \wedge \neg d)] \in \sigma(0)$. So either $\Box\neg d \in \sigma(0)$ and since $\#(\Box = \neg d) < \#(c$ **Until** $d)$ we can use the induction hypothesis to show that $\Box\neg d\,|^{\sigma}_{M} =$ **true** $\Rightarrow c$ **Until** $d\,|^{\sigma}_{M} =$ **false**; or $\neg d$ **Until** $(\neg c \wedge \neg d) \in \sigma(0)$, and by (A4) $\Diamond(\neg c \wedge \neg d) \in \sigma(0)$. Since $\#(\neg c \wedge \neg d) \leqslant \#(c$ **Until** $d)$, $\exists k < i_1$ such that $\neg c \wedge \neg d \in \sigma(i)$. Let i be the smallest such k.

Claim. *$\forall j$, $0 \leqslant j < i$ $d \notin \sigma(j)$ and $\bigcirc(c$ **Until** $d) \notin \sigma(j)$.*

Proof. By induction on j.

Case $j = 0$. Suppose $d \in \sigma(0)$; then by (A6) c **Until** $d \in \sigma(0)$, a contradiction. For the same reason, $c \wedge \bigcirc(c$ **Until** $d) \notin \sigma(0)$. Since $j < i$ and $d \notin \sigma(0)$, $c \in \sigma(0)$, so we must conclude that $\bigcirc(c$ **Until** $d) \notin \sigma(0)$.

Case $j > 0$. Since $i < i_1$, $\sigma(j-1)\,\rho\sigma(j)$. $\bigcirc(c$ **Until** $d) \notin (j-1) \Rightarrow (n' > n+2)$ $\neg\bigcirc(c$ **Until** $d) \in \sigma(j-1) \Rightarrow$ (by (A2)), $\bigcirc\neg c$ **Until** $d \in \sigma(j-1) \Rightarrow \neg c$ **Until** $d \in \sigma(j) \Rightarrow c$ **Until** $d \notin \sigma(j)$, and we conclude as in the case $j = 0$. ▮

Now, since $\#(d) < \#(c$ **Until** $d)$, $\#(\neg c \wedge \neg d) < \#(c$ **Until** $d)$ and $i < i_1$, we can use the induction hypothesis to conclude that $\neg c \wedge \neg d\,|^{\sigma^{+i}}_{M} =$ **true** and $\forall_j$, $0 \leqslant j < i$, $\neg d\,|^{\sigma^{+j}}_{M} =$ **true**. So c **Until** $d\,|^{\sigma}_{M} =$ **false**.

(b) c **Until** $d\,|^{\tau}_{M} =$ **true** $\Leftrightarrow \exists j \in \mathcal{N}$ such that $d\,|^{\tau^{+j}}_{M} =$ **true** and $\forall i < j$ $c\,|^{\tau'^{+i}}_{M} =$ **true** $\Leftrightarrow$ (by the induction hypothesis) $d\,|^{\tau'^{+j}}_{M} =$ **true** and $\forall i < j$ $c\,|^{\tau'^{+i}}_{M} =$ **true** $\Leftrightarrow c$ **Until** $d\,|^{\tau'}_{M} =$ **true**.

$b = \nabla c$. (a) Suppose $\nabla c \in \sigma(0)$.

Let $s =_{\text{def}} \sigma(0)$. We want to show that $\{\sigma \mid \sigma \in P_s,$ $c\,|^{\sigma}_{M} =$ **true**$\}$ is co-m(P_s). By Lemma 9.33 $\{\sigma \mid \sigma \in P_s$, σ is $\#(\Diamond c)$-ultimately standard$\}$ is co-m(P_s). By Lemma 9.36 $\{\sigma \mid \sigma \in P_s$, σ is generic$\}$ is co-m(P_s), so $\{\sigma \mid \sigma \in P_s$, σ is generic and $\#(\Diamond c)$-ultimately standard$\}$ is co-m(P_s). So it is enough to show that $\forall \sigma \in P_s$ such that σ is generic and $\#(\Diamond c)$-ultimately standard, $c\,|^{\sigma}_{M} =$ **true**.

So let $\tau \in P_s$ be a generic ultimately $\#(\Diamond c)$-standard path. We will build a generic $\#(\Diamond c)$-standard path τ' such that $\tau \equiv \tau'$. We will get $\tau(0) \equiv \tau'(0)$, and since $\nabla c \in \tau(0)$, $\nabla c \in \tau'(0)$ (see Lemma 9.11). By A10: $\nabla c \to c$, so $c \in \tau'(0)$. If we use the induction hypothesis part (a), we get $c\,|^{\tau'}_{M} =$ **true**. By the induction hypothesis part (b), $c\,|^{\tau}_{M} = c\,|^{\tau'}_{M}$, so we get $c\,|^{\tau}_{M} =$ **true**.

So now it is left to show τ' as described above. Since τ is $\#(\Diamond c)$-ultimately standard, $\exists j \in \mathcal{N}$ such that τ^{+j}, is $\#(\Diamond c)$-standard. By Lemma 9.13 there are traces s_i, $0 \leqslant i \leqslant j$, such that $s_j = \tau(j)$, $s_i \equiv \tau(i)$, and $s_i \rho s_{i+1}$ $\forall i$, $0 \leqslant i < j$. We will define the path τ' by $\tau'(i) = s_i$ $\forall i$, $0 \leqslant i < j$, $\tau'(i) = \tau(i)$ $\forall i, j \leqslant i$. Clearly, τ' is $\#(\Diamond c)$-standard and $\tau' \equiv \tau$. τ' is generic because τ is generic.

Suppose $\nabla c \notin \sigma(0)$.

We will show that there is a set Q such that $Q \subseteq \{\tau \mid \tau \in P_{\sigma(0)},$ $c\,|^{\sigma}_{M} =$ **false**, and Q is sc($P_{\sigma(0)}$), and so $\nabla c\,|^{\sigma}_{M} =$ **false**. Whether $\sigma(0)$ is terminal or not, there exists (by Lemma 9.12 or Lemma 9.19) a trace E such that $\sigma(0) \equiv E$, $c \notin E$, and if $\sigma(0)$ is terminal, E is terminal and $\sigma(0) \approx E$. By Lemma 9.6, we have E' such that $E\rho E'$. We will define the set Q as follows: $Q =_{\text{def}} \{\tau \mid \tau \in P_{\sigma(o)},$ $\tau(1) = E'$, τ^{+1} is generic and $\#(\Diamond c)$-standard$\}$.

Claim. *Q is* sc($P_{\sigma(0)}$).

Proof. Let $A =_{\text{def}} \{\tau \mid \tau \in P_{E'}$, τ is generic and $\#(\Diamond c)$-standard$\}$, $B =_{\text{def}} \{\tau \mid \tau \in P_{\sigma(0)},$ $\tau(1) = E'\}$. According to Lemma 3.5 there is a natural homeomorphism f between B and $P_{E'}$. Clearly, $f^{-1}(A) = Q$. By Lemma 9.37 A is sc($P_{E'}$), so Q is sc(B). B is open($P_{\sigma(0)}$), so Q is sc($P_{\sigma(0)}$). ▮

Claim. *$\forall \tau \in Q$ $c\,|^{\tau}_{M} =$* **false**.

Proof. Let $\tau \in Q$. Let τ' be the path defined by $\tau'(0) = E$, $\forall i \geqslant 1$, $\tau'(i) = \tau(i)$. τ' is generic $\#(\Diamond c)$-standard path. By the induction hypothesis, part (a), and since $c \notin E$, we have $c\,|^{\tau'}_{M} =$ **false**. But since $\tau' \equiv \tau$, by the induction hypothesis, part (b), $c\,|^{\tau}_{M} =$ **false**. ▮

(b) Let σ be a generic $\#(\Diamond b)$-standard path such that $\tau(0) = \sigma(0)$ (by Lemma 9.37 there exist such σ). Since the truth value of ∇c depends only on the first state of the path, we have $\nabla c\,|^{\tau}_{M} = \nabla c\,|^{\sigma}_{M}$. By part (a), $\nabla c\,|^{\sigma}_{M} =$ **true** $\Leftrightarrow \nabla c \in \sigma(0) \Leftrightarrow \nabla c \in \tau(0)$. With a similar justification, $\nabla c\,|^{\tau'}_{M} =$ **true** $\Leftrightarrow \nabla c \in \tau'(0)$. Since $\tau(0) \equiv \tau'(0)$, $\nabla c \in \tau(0) \Leftrightarrow \nabla c \in \tau'(0)$. ▮

Lemma 9.39. *The model M described above does not satisfy a.*

Proof. By Lemma 9.37, the set A of generic $\#(\Diamond a)$-standard paths beginning at u is of second category. By Lemma 9.38 $\forall \sigma \in A$, $a\,|^{\sigma}_{M} =$ **true** $\leftrightarrow a \in u$. By the way we have built M, $a \notin u$, so $\forall \sigma \in A$ $a\,|^{\sigma}_{M} =$ **false**. So the set $\{\sigma \mid \sigma \in P_u,$ $a\,|^{\sigma}_{M} =$ **true**$\}$ is not co-m(P_u), so M does not satisfy a. ▮

10. EXAMPLES

We will conclude by showing two examples of the use of the logical system presented in this paper.

10.1. Proving Termination Property

First we will use the system presented in this work to describe a simple parallel program and prove that it terminates in a co-meagre set of its possible executions. This program appears in [LPS81] without the proof of its termination. Consider the following two process program illustrated in Fig. 2.

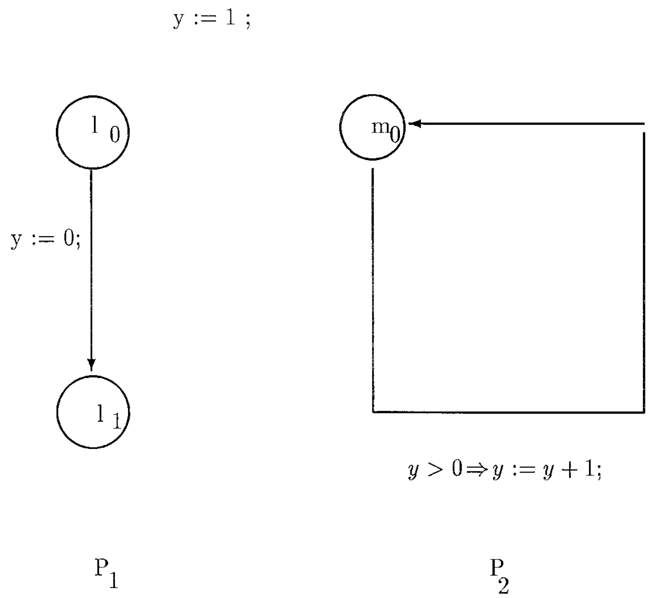

FIGURE 2

The propositional variables we will use are:

atl_0 is **true** iff P_1 is in label l_0.

atl_1 is **true** iff P_1 is in label l_1.

atm_0 is **true** iff P_2 is in label m_0.

$posy$ is **true** iff $y > 0$.

We want to show that the program may terminate in all its possible execution sequences, except a set of executions which is of first category. The program will terminate when P_1 is at label l_1, P_2 is at label m_0, and y is equal to 0. So the claim we want to make about the program may be formalized in the following proposition:

(G) $atl_0 \wedge atm_0 \rightarrow \Diamond\Box(atm_0 \wedge atl_1 \wedge \neg posy)$.

The following propositions will describe the program:

(A) $\Box[(atl_0 \wedge atm_0) \rightarrow [\bigcirc(atl_0 \wedge atm_0) \vee \bigcirc(atl_1 \wedge atm_0)] \wedge \triangle\bigcirc(atl_0 \wedge atm_0) \wedge \triangle\bigcirc(atl_1 \wedge atm_0)]$

This proposition expresses the fact that if P_1 is at label l_0 and P_2 is at label m_0, either P_1 will be activated and will be at label l_1 or P_2 will be activated and will stay at label m_0, and both possibilities can actually occur.

(B) $\Box(atl_1 \rightarrow \Box atl_1)$

This proposition says that once P_1 reaches label l_1, it will always be there.

(C) $\Box atm_0$

This proposition says that P_2 is always at label m_0.

(D) $\Box(atl_0 \vee atl_1)$

P_1 is always either at label l_0 or at label l_1.

(E) $\Box(atl_1 \rightarrow \Box\neg posy)$

Once P_1 is at label atl_1, y will never be bigger than 0.

(F) $\Box\neg(atl_0 \wedge atl_1)$

P_1 cannot be in both label l_0 and label l_1 at the same time.

We can prove that the proposition $A \wedge B \wedge C \wedge D \wedge E \wedge F \rightarrow G$ is valid (see [BE87]).

10.2. Proving Mutual Exlusion and Freedom from Starvation

This example will be a very simple minded example of a mutual exclusion protocol. Each process has its own communication register which can contain only the values 0 and 1. A process can read other processes' registers but it is the only one that is allowed to write in its own register. We assume that the registers are atomic, namely, in any execution, we can assume that the different write and read operations can be assumed to be interleaved (compare with [Lam86]). Besides this assumption, we are not assuming any relation about the relative timing between different processes. An individual protocol is described in Fig. 3. It is not too difficult to see that if any number of processes perform this protocol then the mutual exclusion of the different sections is guaranteed. (If processes A and B are simultaneously in their critical section then each one entered its critical section after writing 1 and then reading all the other communication registers which gave 0. We are referring to the final writing of 1 before the critical section is entered. By our assumption about the atomicity of the registers either the write of A preceded the write of B or vice versa. Without loss of generality assume that the write operation of A preceded that of B. Hence when B was reading the communication register of A it must have read 1, so it would not enter his critical section, hence a contradiction).

The property which is not guaranteed for every execution sequence is the avoidance of starvation or even deadlock. One can easily schedule the actions of two processes trying to get into their critical section such that both keep writing 1, reading the other communication register, and getting 1,

```
$l_1$: $reg[i] = 0$;

$l_2$: Sleep;

$l_3$: $reg[i] = 1$;

$l_4$: If for some $j \neq i$, $1 \leq j \leq n$, $reg[j] = 1$
       then goto $l_1$;

$l_5$: critical section;

$l_6$: goto $l_1$;
```

FIG. 3. The program executed by the ith process.

hence restarting the protocol and so ad infinitum. (It is a general fact that for any protocol such that each process uses only two values, one can not have mutual exclusion and freedom from starvation for *every* execution sequence (see for instance [AM]). On the other hand, one can show that for a co-meagre set of execution sequences deadlock or even starvation is avoided.

Let us describe how a system made of two processes which perform the above protocol can be described in our logic: The two processes will be denoted by A and B. We shall have several propositional variables describing the state of each of this processes, so we shall subscript them by A or B, respectively. So for instance Ci_{A} for $i = 0, 1$ is true if the register of A contains i. Similarly for Ci_{B}. We shall have six propositional variables for each process describing the stage of the protocol it is in (each one has to subscripted by A or B respectively): $\mathit{atl}1, \mathit{atl}2, ..., \mathit{atl}6$. Since we are not making any assumption about the timing (besides the atomicity assumption), each process can stay in the same state arbitrarily long. The atomicity assumption is expressed by the fact that no two "write" or "read" transitions are done simultaneously. The behavior of our system is described as follows (for an axiom where the propositional variables are not subscripted we actually mean the conjunction of the formula where each propositional variable is indexed by A and one which each propositional variable is indexed by B):

Init.

$$\begin{aligned} &\mathit{atl}1 \to \neg(\mathit{atl}2 \vee \mathit{atl}3 \vee \cdots \vee \mathit{atl}6) \\ &\wedge \bigcirc((C0 \textbf{ Until } \mathit{atl}3 \wedge \mathit{atl}2) \vee \mathit{atl}1) \end{aligned}$$

Sleep.

$$\mathit{atl}2 \to \neg(\mathit{atl}3 \vee \mathit{atl}4 \vee \cdots \vee \mathit{atl}6) \wedge \bigcirc(\mathit{atl}2 \vee \mathit{atl}3)$$

This formula intuitively expresses that if the process is in a sleep state (l_2) then it is not in any other state, it has 0 in its register, and in the next moment it can either stay in a sleep state or move to state l_3.

Write.

$$\begin{aligned} &\mathit{atl}3 \to \neg(\mathit{atl}4 \vee \mathit{atl}5 \vee \mathit{atl}6) \\ &\wedge \bigcirc((\mathit{atl}4 \wedge C1 \textbf{ Until } \mathit{atl}1) \vee \mathit{atl}3) \end{aligned}$$

Read.

$$\begin{aligned} &\mathit{atl}4 \to (C0_o \to \bigcirc(\mathit{atl}5 \vee \mathit{atl}4)) \wedge (C1_o \to \bigcirc(\mathit{atl}5 \vee \mathit{atl}4)) \\ &\wedge \neg(\mathit{atl}5 \vee \mathit{atl}6). \end{aligned}$$

In this case we have to refer to the possible transition in terms of the other process, so in the above formula $C0_o$ and $C1_o$ o either A or B and it is the other process than the one for which the formula is stated.

Critical.

$$\mathit{atl}5 \to \bigcirc(\mathit{atl}5 \vee \mathit{atl}6) \wedge \neg \mathit{atl}6$$

Loop.

$$\mathit{atl}6 \to \bigcirc(\mathit{atl}1 \vee \mathit{atl}6)$$

Atomicity. This formula expresses the fact the registers are atomic; that is to say, we do not simultaneously perform read and write transitions on the same register. (The fact that no two reads are done simultaneously follows from the previous formulas, so we have to make sure that A does read the register of B while B is writing it and vice versa.) A read transition in our protocol is characterized by the formula $\mathit{atl}4$; a write transition is identified by $\mathit{atl}1 \vee \mathit{atl}3$.

$$(\mathit{alt}4_{\mathrm{A}} \to \neg(\mathit{atl}1_{\mathrm{B}} \vee \mathit{atl}3_{\mathrm{B}})) \wedge (\mathit{atl}4_{\mathrm{B}} \to \neg(\mathit{atl}1_{\mathrm{A}} \vee \mathit{atl}3_{\mathrm{A}}))$$

The formula we are interested in expresses the fact that if one of our processes is not in a sleeping condition infinitely many times then it will be in the critical section infinitely many times (denote it by NS for "no starvation"):

$$\Box \Diamond \neg \mathit{atl}2 \to \Box \Diamond \mathit{atl}5$$

Another formula expresses the mutual exclusion (ME):

$$\neg(\mathit{alt}5_{\mathrm{A}} \wedge \mathit{atl}5_{\mathrm{B}})$$

The fact that a process behaves according to the protocol and that the registers can have only one value at a time can be expressed by the formula (called correctness)

$$\Box((\mathit{atl}1 \vee \mathit{atl}2 \vee ... \vee \mathit{atl}6) \wedge \neg(C0 \wedge C1))$$

The correctness of the protocol for a co-meagre set of execution sequences (namely, if we start from a situation in which mutual exclusion is not violated, then in the future it will not be violated and neither process will be starved) can be expressed as

$$\begin{aligned} &\nabla[\mathrm{ME} \wedge \mathit{Correct}_{\mathrm{A}} \wedge \mathit{Correct}_{\mathrm{B}} \wedge \mathit{Atomicity}) \\ &\to (\Box \mathrm{ME} \wedge \mathrm{NS})] \end{aligned}$$

The validity of the formula proves the correctness for a co-meagre set of legal execution sequence. We shall not present here the formal proof, which can be rather tedious and can be constructed from the informal argument, but let us just point out that the completeness proof guarantees that such a proof exists.

11. CONCLUSION AND FUTURE WORK

In this paper we formalize the concept of "general execution" of a concurrent program by considering the topology of the space of all possible execution sequences and looking at "co-meagre" sets in this topology. To support this approach, we have shown that for most reasonable notions of fairness used in the literature, the set of all fair executions turns out to be a co-meagre set in the appropriate topology. We have presented a variation of the temporal logic of linear time that can be used for specification and verification of properties that hold in the generic execution of a parallel program. The formal system was already suggested by Lehmann and Shelah [LS82], but we give it different, topological semantics. Our system is decidable, sound, and complete for models of arbitrary size, but it has the finite model property.

In [FHLdR79], and also in some more recent work [Var85, KP90], it has been suggested to distinguish between non-determinism which is external to the program and caused, for example, by scheduling or relative timing of different processes and non-determinism which is internal to a particular process. Fairness can be applied to both types of non-determinism (see for example [AFK88]). Vardi [Var85] suggests an analysis of distributed probabilistic messages, based on the notion of *concurrent stochastic* process, which partition the state space of the system into two types of states: states in which internal non-deterministic choices are taken, and states in which external non-deterministic choices are take. These two types of states alternate during the execution of the program. A *schedule* is a subgraph of the execution tree obtained by selecting one branch out of each set of external non-deterministic choices but keeping the full sets of internal non-deterministic choices. Hence it is possible to extract many schedules from the space of all possible executions of a program. In Vardi's approach internal non-determinism is expressed by probabilistic choices, and each choice can be analyzed as a Markov process. A property holds (probabilistically) over a concurrent stochastic process if it holds with probability 1 over every schedule. This approach considers the worst case for external non-determinism, and for internal choices it is ready to ignore a set of executions of measure 0 in each schedule.

The logical system presented in this paper cannot be used to prove properties of programs which depend on the above distinction between internal and external nondeterminism. The system presented by Lehmann and Shelah is also not capable of handling such variants. However, our language can be modified to handle such a refined analysis. One should generalize the operator ∇ to a binary operator $\nabla(\phi, \psi)$ having the following meaning: consider a model $M = \langle S, u, l, R \rangle$ as in Definition 3.1. A ϕ-determined submodel of M (notation: M_ϕ) is a model $\langle S_\phi, u, l', R' \rangle$, where the following conditions hold:

1. $S_\phi \subseteq S$;
2. For every $s \in S_\phi$, if ϕ is satisfied by the model $\langle S, s, l, R \rangle$, then $R'(s, t) \Rightarrow (R(s, t)$ and there is no $v \neq t$ such that $R'(s, v))$;
3. For each $s \in S_\phi$, $l'(s) = l(s)$.

Informally, to get M_ϕ we prune the original tree (M) in such a way that each state satisfying ϕ will have exactly one child. We then say that $\nabla(\phi, \psi)$ holds at a state s in M if for *every* ϕ-determined submodel of M starting at s, for a co-meagre set of paths ψ holds. Our unary operator $\nabla\psi$ is a special case of the new binary operator by taking $\phi = \mathbf{false}$. Note that taking $\phi = \mathbf{true}$ means "for every branch ψ holds," so the new language covers also the expressibility of the temporal logic of branching time.

The two variants of non-determinism mentioned above can easily be handled in this language by having a propositional variable p which intuitively means that we are in a state where internal non-deterministic choices are taken. The statement that ψ holds in "almost all" executions (no matter what are the external choices) can be expressed by $\nabla(\neg p, \psi)$. We are currently working on finding a complete axiom system for this generalized language.

APPENDIX A. A COUNTER EXAMPLE

In this appendix we give an outline of a counter example to Lemma 24 which appears on p. 194 of [LS82].

The lemma is stated as follows:

LEMMA 24. *Let $b \in \Gamma_n$, σ be an $\Omega(b)$-standard generic path of U, and τ and τ' be two equivalent sequences of U; then*

(a) $b \mid_U^\sigma = \mathbf{true} \Leftrightarrow b \in \sigma(0)$

(b) $b \mid_U^\tau = \mathbf{true} \Leftrightarrow b \mid_U^{\tau'} = \mathbf{true}$.

(Compare the above lemma to Lemma 9.38 in this paper.)

Since we take this lemma out of context, we should mention that U is the model constructed in the completeness proof and Ω is a function from the set of formulas to $\mathcal{N}$, which assigns a fixed natural number to every formula, in a way explained in Sect. 4 of [LS82]. [LS82] define an m-standard sequence as follows (see third paragraph on p. 194 of [LS82]):

> Let m be a natural number. Let σ be a sequence of states of U. If there exists a $n \in \mathcal{N}$ such that $\sigma(n)$ is terminal, for all i such that $0 \leqslant i < n + m$ we have $\sigma(i) \rho \sigma(i+1)$, and for all i such that $n + m \leqslant i$ we have $\sigma(i) \approx \sigma(i+1)$, we shall say that σ is a m-standard sequence.

The reader has probably noticed that our definition of an m-standard sequence is substantially different. The reason is that as we prove in the sequel, with the above definition of m-standardness, for every natural number assigned by the function Ω to the formula b, we can find a counterexample to Lemma 24, as stated in [LS82]. We will show that for every m, we can build a model M and an m-standard generic path σ (in the sense of [LS82]) such that p **Until** $q \notin \sigma(0)$ but p **Until** $q \mid_M^\sigma =$ **true**, where $p, q \in Pvar$.

Suppose $Pvar = \{p, q\}$ and consider the following model M: M has three states: $s0$, $s1$, $s2$. At $s0$, $p =$ **true**, $q =$ **false**, at $s1$ both p and q are **true** and at $s2$ both p and q are **false**. The transitions are as follows: from $s0$ we can move to $s1$ or to $s2$ or stay at $s0$, from $s1$ we can move to $s0$, and from $s2$ we can move to $s0$.

Now we construct a legal sequence σ^* in this model: We start from $s0$ and we require it to have the following property: If at a certain point k and for some formula a, the set of paths starting from the state at k and satisfying a, is of first category, then the tail of σ^*, avoids this set. (You do it by induction; At each stage we get a countable list of nowhere dense sets we are required to avoid. They are made up of previous sets we did not run away from, as well as new set, because we have new k's. Now we pick one of these sets and make a finite extension of σ^* so as to avoid this set. We can make these "pickings" in such a way that every set we are required to avoid will eventually be picked.)

Let T_n be the theory made up of all statements holding at the n-th point of σ^*. Clearly, for every n T_n is complete. Note that in view of the construction of σ^*, we have that if ∇a holds at n then a is in T_n, hence T_n is consistent.

Now say that a point n of σ^* is *m-standard of the first kind* if it is the initial point of a sequence that looks like $s0, s0, s0, \ldots$ $(m+1$ times$)$, $s1$, $s0$. It is an *m-standard point of the second kind* if it looks like $s0, s0, \ldots$ $(m+1$ times$)$, $s2$, $s0$. Note that by the construction of our sequence it has (for every m) infinitely many m-standard points of both kinds. (The complement is a set of paths described by a formula which is satisfied only by a first category set; hence we have avoided it).

A complete consistent theory is said to be *good* for a particular infinite set of points of our sequence if every formula in it belongs to infinitely many T_n, such that n is in the set.

The following facts are rather easily verified:

- For every infinite set of points there is a good theory for this set. (*Proof.* Let T be the set of sentences which are true at almost all the points of the given infinite set (namely, except possibly finitely many points). Using the particular way we have constructed σ^*, we can easily show that T is consistent, Hence there is an extension of T to a complete consistent theory T^*. T^* is as required because if a sentence a is in T^*, it must be in infinitely many T_n's from our set, because otherwise its negation is in T, and a cannot be in T^*).
- A theory good for an infinite set is terminal.
- If $T1$ and $T2$ are theories good for two (possibly different) infinite sets then $T1 \leqslant T2$ and $T2 \leqslant T1$.
- If $T1$ and $T2$ are good for two (possibly different) infinite sets A, B such that at every point of A and at every point of B σ^* has the same state (and it is the same for A and B!) then $T1$ is an alternative to $T2$. (*Proof.* Assume ∇a is in $T1$. Pick n in A such that ∇a is in T_n. Hence the set of paths starting from s (s is the common state at every point of A and B) satisfying $\neg a$ is of first category. Therefore by the construction of σ^*, at every point of B also the tail of σ^* does not satisfy $\neg a$, so a is in T_m for every m in B; hence a is in $T2$.)

Note that the same argument shows that $T1^{+k}$ is an alternative for $T2^{+k}$, where T^{+k} is the k-th successor of T.

Now fix m. Let $T1$ be a theory good for the set of m-standard points of the first kind and $T2$ a theory good for the set of m-standard points of the second kind. Recall that $T1$ and $T2$ are terminals. We now construct a sequence τ in the transition system constructed in [LS82] which will be a counterexample to their Lemma 24.

The sequence τ will be as follows: trace($T2$), trace($T2^{+1}$), trace($T2^{+2}$), trace($T2^{+m}$), trace($T1^{+(m+1)}$) and from there on any continuation that will make the sequence generic (e.g., since $\approx$ is reflexive and ρ is a function we can continue with trace($T1^{+(m+2)}$), trace($T1^{+(m+3)}$)...). Note that τ is m-standard in the sense of [LS82]. τ satisfies p **Until** q because we moved at the $(m+1)$-th point to trace($T1^{+(m+1)}$), but trace($T2$), its first state does not contain p **Until** q because $T2$ is good for the points of the second kind.

Since m was arbitrary it means that no fixed length of "standardness" is sufficient to guarantee the truth of Lemma 24.

The same modified definition that we used for *standard paths* (Definition 9.28) will correct the error of [LS82] for the probabilistic (measure theoretic) case.

ACKNOWLEDGMENTS

We thank Nissim Frances and Shmuel Katz for useful discussions. The anonymous referees had very helpful comments and suggestions.

This work was supported in part by Grant AFOSR 90-0136.

Received November 26, 1990; final manuscript received July 7, 1995

REFERENCES

[AFK88] Apt, K. R., Frances, N., and Katz, S. (1988), Appraising fairness in languages for distributed programming, *Distrib. Computing*. **2**, 226–241.

[AM] Abraham, U., and Magidor, M., The mutual exclusion problem—The quest for a minimal solution, submitted for publication.

[AN80] Arnold, A., and Nivat, M. (1980), Metric interpretations of infinite trees and semantics of non-deterministic programs, *Theor. Comput. Sc.* **11**, 181–225.

[AS85] Alpern, B., and Schneider, F. B. (1985), Defining livenes. *Inform. Process. Lett.* **21**, 181–185.

[BE87] Ben-Eliyahu, R. (1987), "Proving Correctness for the General Execution." Master's thesis, Mathematics and Computer Science Institute, The Hebrew University, Jerusalem.

[BZ82] Bakker, J. W., and Zucker, J. I. (1982), Processes and the denotational semantics of concurrency, *Inform. and Control* **54**, 70–120.

[CLP83] Cohen, S., Lehmann, D., and Pnueli, A. (1983), Symmetric and economical solutions to the mutual exclusion problem in a distributed system, *in* "Proceedings of the 10th International Colloquium on Automata, Languages and Programming, Barcelona, Spain, July 1983" (J. Diaz, Ed.), pp. 128–136, Springer-Verlag, Berlin/New York.

[Dix84] Dixmier, J. (1984), "General Topology." Springer-Verlag, New York.

[FHLdR79] Francez, N., Hoare, C. A. R., Lehmann, D. J., and de Roever, W. P. (1979), Semantics of nondeterminism, concurrency, and communication, *J. Comput. System Sci.* **19**, 290–308.

[Fra86] Francez, N. (1986), "Fairness," Springer-Verlag, Berlin/New York.

[GPSS80] Gabbay, D., Pnueli, A., Shelah, S., and Stavi, J. (1980), On the temporal analysis of fairness, *in* "Conference Record of the 7th Annual ACM Symposium on Principles of Programming, Languages, Las Vegas, Nevada," pp. 163–173.

[Kel55] Kelley, J. L. (1955), "General Topology," Van Nostrand, 1995.

[KP90] Katz, S., and Peled, D. (1990), Interleaving set temporal logic, *Theoret. Comput. Sci.* **75**, 263–287.

[Lam86] Lamport, L. (1986), The mutual exclusion problem. 1. A theory of interprocess communication, *J. Asoc. Comput. Mach.* **33** (2), 313–326.

[LPS81] Lehmann, D., Pnueli, A., and Stavi, J. (1981), Impartiality, justice, and fairness: The ethics of concurrent termination, *in* "Proceedings of the 8th International Colloquium on Automata, Languages, and Programming, Acco, Israel," pp. 264–277.

[LR81] Lehmann, D., and Rabin, M. (1981), On the advantages of free choice: A symmetric and fully distributed solution to the dining philosophers problem, *in* "Conference Records of the Annual ACM Symposium on Principles of Programming Languages, Williamsburg, VA," pp. 133–138.

[LS82] Lehmann, D., and Shelah, S. (1982), Reasoning with time and chance, *Informat. and Control* **53**, 165–198.

[Pnu83] Pnueli, A. (1983), On the extremely fair treatment of probabilistic algorithms. *in* "STOC-83: Proceedings of the 15th ACM Symposium on Theory of Computing," pp. 278–290.

[Var85] Vardi, M. Y. (1985), Automatic verification of probabilistic concurrent finite-state programs, *in* "FOCS-85: Proceedings of the 26th IEEE Symposium on Foundations of Computer Science," pp. 327–338, Comput. Soc. Press.