Probabilistic Temporal Logics for Finite and Bounded Models

Sergiu Hart and Micha Sharir(*)

School of Mathematical Sciences
Tel Aviv University
Ramat Aviv, Tel Aviv 69978, ISRAEL

ABSTRACT

We present two (closely-related) propositional probabilistic temporal logics based on temporal logics of branching time as introduced by Ben-Ari, Pnueli and Manna and by Clarke and Emerson. The first logic, PTL_f, is interpreted over finite models, while the second logic, PTL_b, which is an extension of the first one, is interpreted over infinite models with transition probabilities bounded away from 0. The logic PTL_f allows us to reason about finite-state sequential probabilistic programs, and the logic PTL_b allows us to reason about (finite-state) concurrent probabilistic programs, without any explicit reference to the actual values of their state-transition probabilities. A generalization of the tableau method yields exponential-time decision procedures for our logics, and complete axiomatizations of them are given. Several meta-results, including the absence of a finite-model property for PTL_b, and the connection between satisfiable formulae of PTL_b and finite state concurrent probabilistic programs, are also discussed.

(*) Work by the second author has been supported in part by a grant from the Bat-Sheva Fund and by a grant from the U.S.-Israeli Binational Science Foundation.

1. Introduction

Recent progress in the theory of probabilistic programs [SPH], [HSP], [HS] has yielded relatively simple methods for verification of certain properties of such programs. Sequential probabilistic programs have been represented in [SPH] as discrete Markov chains, whereas concurrent probabilistic programs have been represented in [HSP] and [HS] as processes involving cooperation of several Markov chains (with a common state space) obeying certain "fairness" constraints. In both cases, if one assumes that the state space of the programs in question is finite, then one can obtain simple algorithmic techniques for analyzing and proving *termination* of such programs. For sequential programs these techniques are essentially classical results in Markov chain theory, whereas for concurrent programs new techniques had to be developed. In both cases, the actual values of the state-transition probabilities proved to be irrelevant for the properties in question.

These encouraging results have motivated the study of logics for probabilistic programs, as presented in this paper. These logics are expressive enough to allow one to express various properties of such programs, including invariant and liveness properties, without explicit reference to the values of the transition probabilities. The first logic, which we call PTL_f, is intended for reasoning about sequential programs, whereas the second logic, called PTL_b, extends the first one and is intended for reasoning about concurrent programs. Both logics are based (at least syntactically) on existing temporal logics for branching time [BPM], [CE]. These logics are interpreted over models which can simulate the execution of probabilistic programs; for PTL_f these are essentially finite Markov chains, whereas for PTL_b they are infinite stochastic processes whose state-transition probabilities are bounded away from 0 (this assumption holds for finite-state concurrent probabilistic programs since there are only finitely many different state-transitions).

It turns out that satisfiability of formulae in both logics is decidable, in one-exponential time, by decision procedures based on the tableau technique which generalize similar procedures for the nonprobabilistic logics of [BPM] and [CE]. The probabilistic context of our logics makes these procedures more complicated than their nonprobabilistic counterparts, and introduces into them some special techniques which are variants of the techniques used in [HSP] for analyzing termination of concurrent probabilistic programs.

Together with these decision procedures, we also provide complete axiomatizations for both logics, and show that the same decision procedures can be used to construct a proof of the negation of any unsatisfiable formula.

Moreover, by inspection of the decision procedure for PTL_b, we see that for many (satisfiable) formulae of that logic the model constructed by that procedure can be replaced by a finite model. This establishes a connection between satisfiability of a formula in PTL_b and its satisfiability in PTL_f, when certain conditions hold. In any case, the model constructed by the decision procedure can

always be viewed as the execution tree of some finite-state concurrent probabilistic program, under some schedule. Some additional properties of the models of formulae in these logics are also discussed.

Probabilistic logics of various sorts have been recently proposed by various authors [FH],[Fe],[Pn],[KL],[LS],[Ko]. Some of these logics pertain only to sequential programs and involve explicit reference to the values of transition probabilities. Like our logic PTL_b, the logics proposed by Pnueli [Pn] and by Lehmann and Shelach [LS] also aim to reason about concurrent probabilistic programs and do not refer explicitly to the values of probabilities involved. However, the logic of Pnueli is not complete; the logic of Lehmann and Shelach is more expressive than ours, and consequently the presently available decision procedures for that logic are much more inefficient than ours (the best such procedure, given in [KL], runs in doubly exponential nondeterministic time). Both these logics are based on temporal logic of linear time, which, in our opinion, is somewhat inappropriate, since it does not fully correspond to the more standard branching tree-like model for the execution of probabilistic programs. Use of temporal logic of branching time is a more appropriate choice, and it makes the logic and its interpretation much more natural.

The paper is organized as follows. In Section 2 we define the syntax and semantics of our logics, and make a few basic observations concerning these notions. In Section 3 we give axiomatic systems for both logics. Section 4 describes the decision procedures for our logics, shows how to construct a model for a satisfiable formula in either logic. Section 5 establishes the completeness of the axiomatic systems, in the sense that the proof of any formula p for which $\sim p$ is unsatisfiable, can be mechanically obtained from the tableau constructed for $\sim p$. Section 6 discusses some meta-results concerning properties of formulae and their models.

2. Syntax and Semantics

Our two logic systems, denoted PTL_f and PTL_b, are almost identical syntactically, but differ in the interpretation of their formulae. Like the (nonprobabilistic) logic *CTL*, our logics are based on the propositional calculus extended by three modal operators, two unary prefix operators $\forall X$ and $\forall F$, and one binary infix operator $\forall U$. The operator $\forall F$ in PTL_f is redundant, and can be defined in terms of the other two operators; however, this is not the case in PTL_b.

PTL_f and PTL_b are interpreted as follows. A *model* of PTL_b is a (discrete) Markov chain, possibly having infinitely many states, for which there exists $\alpha>0$ such that all nonzero transition probabilities of the chain are $\geq\alpha$; and a specified initial state. In addition, there is an assignment of truth values to all propositions appearing in a given formula at each state of the chain. Formally, such a model M is defined as a quadruple (S,P,s_0,ρ), where S is a set of states, $s_0 \in S$ is the initial state, P is a transition probability matrix (i.e. mapping on $S\times S$ with $\sum_{t \in S} P(s,t) = 1$ for all $s \in S$) each of whose entries is either 0 or $\geq\alpha$, where α is some positive constant, and ρ is a mapping on S assigning to each $s \in S$ the set of true propositions at that state. For convenience, we abbreviate $p \in \rho(s)$ as $p \in s$.

A model of PTL_f is defined similarly, with the additional requirement that the set S be *finite* (the requirement of the boundedness of the transition probabilities clearly holds here).

Each model M induces a probability measure μ_M on the space Ω_M of all infinite paths in S starting at s_0. Validity of a formula p of either logic, in an appropriate model M, denoted $\models_M p$, is defined recursively as follows.

(i) If p is a proposition, then $\models_M p \Leftrightarrow p \in s_0$.

(ii) If $p=\sim q$, then $\models_M p \Leftrightarrow \not\models_M p$.

(iii) If $p=q\vee r$, then $\models_M p \Leftrightarrow \models_M q$ or $\models_M r$, and similarly for all other logical connectives.

(iv) If $p=\forall Xq$, then $\models_M p \Leftrightarrow \models_{M_1} p$ for all $s_1 \in S$ such that $P(s_0,s_1) > 0$, where M_1 is the model M with initial state s_1, instead of s_0.

(v) If $p=q\forall Ur$, then $\models_M p \Leftrightarrow \mu_M(A_{q,r})=1$, where

$$A_{q,r} = \{\omega=(s_n) \in \Omega_M \mid inf\{n \mid \not\models_{M_n} q\} \geq inf\{n \mid \models_{M_n} r\}\},$$

and M_n is the model M with initial state s_n instead of s_0. I.e. $A_{q,r}$ consists of paths along which either q always holds, or else q holds until the first time r holds.

(vi) If $p=\forall Fq$, then $\models_M p$ iff there exists a stopping time N on Ω_M which is μ_M-almost surely finite, such that $\models_{M_N} q$ for each ω with $N(\omega)<\infty$. (A *stopping time* N is a mapping from Ω_M into $\{0,1,2,\ldots,\infty\}$, such that $N(\omega) = n$ implies $N(\omega') = N(\omega)$ for each path ω' which coincides with ω at all steps up to, and including n. Thus, N may depend only on $s_0,s_1,\ldots,s_{N(\omega)}$ (the past and the present), but not on $s_{N+1},\ldots$ (the future)).

Remarks:

(1) The negations of the modal operators $\forall X$, $\forall U$ and $\forall F$ are defined as follows:

$$\exists Xp \equiv \sim(\forall X \sim p).$$

$$p\exists Uq \equiv \sim((\sim q)\forall U(\sim p)).$$

$$\exists Gp \equiv \sim\forall F\sim p$$

(2) Intuitively, $\models_M \forall X p$ means that p holds at all immediate successors (sons) of the initial state of M. Similarly, $\models_M \exists X p$ means that p is valid in at least one son of the initial state of M. $\models_M p\forall Uq$ means that along all paths ω starting at s_0 and consisting only of transitions with nonzero probability, p holds at all states of ω up to the first state, if any, at which q holds. Similarly, $\models_M p\exists Uq$ if there exists a *finite* path ω of states reachable from s_0 via transitions having nonzero probabilities, such that p holds at all states of ω, and q holds at the final state of ω. $\models_M \forall Fp$ if p is valid eventually on almost every path. Also, $\models_M \exists Gp$ if the μ_M-measure of the set of paths along which p always holds (i.e. the set $A_{p,false}$), is positive.

(3) The modal operators $\forall Gp$ and $\exists Fp$ of [BPM] can be defined in both PTL_b and PTL_f in the following usual way:

$$\exists\forall F p = \text{true}\ \exists\forall U\ p$$

$$\forall G p = p\ \forall U\ \text{false}$$

(4) In PTL_f, the operator $\forall F$ and its negation $\exists G$ can be eliminated from the logic, by defining them in terms of the other operators, as follows:

$$\exists G p = p\ \exists\forall U\ (\forall G p) = p\ \exists\forall U\ (p\ \forall U\ \text{false})$$

$$\forall F p = (\exists\forall F p)\ \forall U\ p = (\text{true}\ \exists\forall U\ p)\ \forall U\ p$$

These two definitions are quite nonobvious, and are special to the finite model interpretation of PTL_f. $\models_M \forall F p$ if on μ_M-almost every path p holds eventually. However, in the case of finite models (i.e. finite Markov chains), this is equivalent to requiring that on every path ω and every state s_n along ω before the first time (if any) p holds on ω, there exists a path from s_n on which p eventually holds. This latter property is always implied by the interpretation of $\models_M \forall F p$ as defined above (including infinite Markov chains); the reverse implication can be proved for *finite* Markov chains using standard "0-1 law" arguments, similar to those in [HSP]. The definition of the operator $\exists G p$ follows by negation. However, in PTL_b there is no way to define these operators in terms of the other operators, so that the core of PTL_b will have to include also the operator $\forall F$. For uniformity of notation, we will include $\forall F$ also as a basic operator of PTL_f, and use the above definition of $\forall F$ as an axiom.

(5) The intended application of the logic PTL_f (resp. of PTL_b) is to reason about *finite−state* probabilistic sequential (resp. concurrent) programs. It is easily seen that each possible execution of such a program can be represented as a model of the corresponding logic, such that the model states are program states, and the propositions contained in each such state are properties of the state. In this perspective, it is noteworthy that the behavior of a finite state concurrent program (involving more than one process) cannot always be modeled by a finite Markov chain. For example, suppose that the program consists of three states s_1, s_2, and s_3, and of two processes k_1, k_2, such that under k_1 the transitions having nonzero probability are (s_1,s_1), (s_1,s_2), (s_2,s_1), (s_2,s_2), (s_3,s_3), and under k_2 they are (s_1,s_3), (s_2,s_3), (s_3,s_3). Consider the schedule σ starting at s_1 and defined by the rule: schedule k_1 repeatedly until the first time at which the number of visits at s_2 is greater than or equal to the number of visits at s_1; in this case schedule k_2, and then schedule k_1 and k_2 alternately. Then the execution of the program under σ cannot be modeled by a finite-state Markov chain; in fact this execution is identical to the behavior of a random walk on 0,1,2,..., with absorption at 0. Moreover, the fairness of σ *does* depend on the transition probabilities of k_1 at s_1 and s_2 (e.g. σ is fair if $P^{k_1}_{s_1,s_2}, P^{k_1}_{s_2,s_2} \geq \frac{1}{2}$, and is unfair if both these probabilities are $< \frac{1}{2}$.)

The problem with this σ is that it is not *finitary*. Roughly speaking, a schedule is finitary if it is a finite automaton, whose decisions are based only on checking whether the past execution history belongs to one of several regular languages. It is easily seen that for finite-state programs, their execution under a schedule σ can be modeled by a finite-state Markov chain if and only if σ is finitary.

Fair schedules need not be finitary. However, it can be shown from Theorem 1.1 of [HSP] that almost-sure termination by any fair schedule can be effectively decided by essentially considering only finitary (fair) schedules, and therefore is a property that can be stated and verified in PTL_f. This interplay between PTL_b and PTL_f will be studied in more generality in Section 6 below. We will obtain there the property just noted as a special case of a more general rule, which gives sufficient conditions for formulae of PTL_b to be equivalently represented in PTL_f.

3. Axiomatic Systems

The following axiomatizations of PTL_f and of PTL_b are shown to be complete: These axiomatizations include all axioms and inference rules of the propositional calculus, plus the following additional axioms and rules (some of which are similar to the axioms and rules of *UB* [BPM], while others are special to our logic, and are required to handle the probabilistic nature of our models):

Axioms and Rules Common to both Logics:

Axioms:

(A1) $\forall X(p \supset q) \supset (\forall X p \supset \forall X q)$

(A2) $p\ \forall U\ q \supset q \vee (p \wedge \forall X(p\ \forall U\ q))$

Inference rules:

(R1) $\vdash p \Rightarrow \vdash \forall X p$

(R2) $\vdash r \supset q \vee (p \wedge \forall X r) \Rightarrow \vdash r \supset p \forall U q$

(R3) $\vdash p \Rightarrow \vdash \sim(\forall F \sim p)$

Additional Axioms for PTL_f

(A3) $\forall F p \equiv (\text{true}\ \exists\forall U\ p)\ \forall U\ p$

Additional Axioms and Rules for PTL_b

(A4) $\forall F p \equiv p \vee \forall X \forall F p$

(A5) $\forall F \forall F p \supset \forall F p$

(A6) $(p \forall U q) \wedge \forall F(\sim p) \supset \forall F q$

(R4) $\vdash r \supset p \vee (\forall X \forall F r \wedge \exists\forall X p) \Rightarrow \vdash r \supset \forall F p$

Let us comment briefly on the interpretations of these axioms and rules in our logics, so as to justify their soundness. Axioms (A1) and (A2) and rules (R1) and (R2) are nonprobabilistic and are sound in our interpretations, as well as in the interpretations of the nonproba-

bilistic logic *CTL*. (It has been pointed out by Amir Pnueli that these axioms and rules can be used to simplify existing axiomatizations for that logic.) The axiom (A1) and the rule (R1) are taken from [BPM], and their soundness follows from the definition of $\forall X$, as in [BPM]. Axiom (A2) states that $r = p \forall U q$ satisfies the implication $r \supset q \vee (p \wedge \forall X r)$, which again is obvious from the definitions of the operators $\forall U$ and $\forall X$, whereas rule (R2) states that $p \forall U q$ is the "largest" solution to that implication, in the sense that it is implied by any other solution. The soundness of this rule can be proved by a simple inductive argument.

The soundness of the rule (R3) is also easy to establish from the definitions. As noted in the preceding section, axiom (A3) is sound only under finite-model interpretations; this can be shown using standard "zero-one" arguments, as e.g. those given in Theorem 2.2 of [HSP].

The axioms (A4)-(A6) of PTL_b are all probabilistic, and state various properties of almost-surely finite stopping times. (A4) states that an event p happens eventually almost surely if and only if it either happens now, or else it happens eventually almost surely from any next instance on. Axiom (A5) states that if there exists an almost surely finite stopping time N such that for each path ω with $N = N(\omega) < \infty$, the event p will happen eventually almost surely after reaching ω_N, then p will happen eventually almost surely. (In other words, the composition of a family of almost-surely finite stopping times on an almost-surely finite stopping time yields an almost-surely finite stopping time.) Axiom (A6) states that if p holds continuously until the first time (if any) at which q holds, and if there exists an almost surely finite stopping time N at which p does not hold, then there exists another almost surely finite stopping time $N' \leq N$ at which q holds. The soundness of this axiom is immediate from the definitions.

Remark: Axioms (A4)-(A6) are not specific to the bounded-model interpretation of PTL_b, but rather hold also in general (unbounded) models, as can be easily checked.

Finally, the rule (R4) states that for r to imply that p will eventually hold almost surely, it is sufficient to require that r implies that either p holds now, or that at least one succeeding state satisfies p, and at the same time r will hold once more eventually after every succeeding state. To prove the soundness of this rule, we argue as follows. Let S_0 denote the set of all states s in S at which r holds, and which are reachable from the initial state s_0 via paths along which p did not hold yet (except possibly at s itself). Assume $s_0 \in S_0$ (for otherwise there is nothing to prove). For each $s \in S_0$ let β_s denote the probability that p will hold eventually, given that we have reached s. The premise of (R4) implies that $\beta_s \geq \alpha$, for each $s \in S_0$. Let $\gamma = \inf_{s \in S_0} \beta_s \geq \alpha$. Assuming that $\gamma < 1$, let $s \in S_0$ be such that $\beta_s < \gamma + \frac{\alpha(1-\gamma)}{2}$. Again, the premise of (R4) implies that

$$\gamma + \frac{\alpha(1-\gamma)}{2} > \beta_s \geq \alpha + (1-\alpha)\gamma$$

which is plainly impossible. Hence $\gamma = 1$, so that in particular $\beta_{s_0} = 1$, which is what we wanted to show.

Remark: It would be tempting to replace (R4) by the simpler sound rule

(R4') $\quad \vdash r \supset p \vee (\forall X r \wedge \exists X p) \Rightarrow \vdash r \supset \forall F p$

However, the resulting axiomatic system will not be complete. In fact, the following formula w, which is a variant of (R4),

$$r \wedge \Big(r \supset p \vee (\forall X \forall F r \wedge \exists X p)\Big) \forall U \textbf{ false} \supset \forall F p$$

is not provable from the modified axiomatic system. To see this, consider the interpretation of these axioms under models M which are defined as in PTL_b, except that their associated transition probabilities are not required to be bounded away from 0. Instead we require that for each state s at the i-th level of M we have $P(s,t) \geq \frac{1}{i}$, for each nonzero transition probability $P(s,t)$. It is easy to see that all axioms and rules of the modified system are sound under this interpretation. Indeed, everythind except (R4') is either nonprobabilistic or holds in general unbounded models. Concerning (R4'), suppose that a model M satisfies the premise of (R4'), and that r holds at the initial state of M. Then the probability that p still does not hold after n levels of M is at most $\prod_{i=1}^{n}(1-\frac{1}{i}) \to 0$ as $n \to \infty$. Nevertheless, it is easy to construct a model M of this new kind which does not satisfy the variant w of (R4) given above. To obtain M, take a sequence $\{i_t\}$ of levels for which $\prod_{t=1}^{\infty}(1-\frac{1}{i_t}) > 0$, and define M so that r holds at the root and at each node in each of the levels i_t. For each node n at the i_t level, p holds at exactly one son m of n, with $P(n,m) = \frac{1}{i_t}$. It is then easy to check that M satisifes the precedent of w but not its consequent.

Hence, the "essence" of the bounded-model interpretation of PTL_b is captured by the rule (R4).

Theorems Common to PTL_f and to PTL_b

We next list a few theorems provable from the core set of axioms and inference rules common to both logics, most of which are needed in subsequent sections.

(T1) $\forall X(p \wedge q) = \forall X p \wedge \forall X q$

(T2) $\forall X p \vee \forall X q \supset \forall X(p \vee q)$

(T3) $p \forall U q = q \vee (p \wedge \forall X(p \forall U q))$

(T4) $p \forall U q \wedge ((\sim q) \forall U r) \supset p \forall U r$

(T5) $(p \vee q) \forall U r \supset p \forall U (q \vee r)$

(T6) $(p \forall U r) \wedge (q \forall U r) = (p \wedge q) \forall U r$

(T7) $(p \supset \forall X p) \forall U q \supset (p \supset (p \forall U q))$

(T8) $\forall F p \supset \textbf{true} \exists U p$

(T9) $\forall X p \supset \forall F p$

We can also deduce a few additional inference rules:

(R1') $\quad \vdash p \supset q \Rightarrow \vdash \forall X p \supset \forall X q$

(R2') $\quad \vdash p \Rightarrow \vdash p \forall U q$ for any q

(R5) $\quad \vdash p \Rightarrow \vdash \forall F p$

(R6) $\quad \vdash p \supset q \Rightarrow \vdash \forall F p \supset \forall F q$

(R7) $\quad \vdash r \supset \forall F(p \vee (\forall X \forall F r \wedge \exists X p)) \Rightarrow \vdash r \supset \forall F p$

Proofs of these theorems and rules are omitted in this version.

4. The Tableau Method

In this section we modify the tableau method described in [BPM] to obtain exponential-time decision procedures for formulae in PTL_f and in PTL_b under the respective finite-model and bounded-model interpretations. It suffices to treat the (more complicated) case of PTL_b, since the tableau construction in PTL_f can be obtained as a special case (using the formulae given in Remark (4) of Section 2). The tableau construction for our logics is similar to that of [BPM] in many details, but differs from it in several significant aspects, reflecting the probabilistic context of our interpretations.

Given a formula p_0 of PTL_b which we wish to test for satisfiability, we construct from it a finite directed graph T, called *tableau*, each of whose nodes n is labeled by a set F_n of formulae (intuitively, formulae to be fulfilled at n), some of which have already been "expanded", while others are still "unexpanded". Initially T contains a single node n_0 (the root), and $F_{n_0} = \{p_0\}$, with p_0 unexpanded. T is then constructed inductively as follows. At each step we pick a node n having no successors, and a formula $p \in F_n$ which has not yet been expanded. We then expand p by one of the rules stated below, thereby creating outgoing edges from n, some of which may lead to newly created nodes of T, while others may point back at nodes already present in T.

Let n be a node of T. It can be expanded either by an α expansion, a β expansion or an X expansion. An α expansion is obtained by picking an unexpanded formula r of F_n having one of the forms in the first column of Table 1, creating one son n_1 of n and putting $F_{n_1} = F_n \cup \{r_1, r_2, \cdots\}$, where $r_1, r_2, \cdots$ are the corresponding formulae in the other columns of the table. Similarly, a β expansion is obtained by picking an unexpanded formula r of F_n having one of the forms in the first column of Table 2, creating two sons n_1 and n_2 of n, and putting $F_{n_1} = F_n \cup \{r_1\}$; $F_{n_2} = F_n \cup \{r_2\}$.

r	r_1	r_2
$p \vee q$	p	q
$p \forall u q$	q	$p \wedge \forall X (p \forall u q)$
$p \exists u q$	$p \wedge q$	$p \wedge \exists X (p \exists u q)$
$\forall F p$	p	$\forall X \forall F p \wedge \exists X \forall F p$
$\sim(p \wedge q)$	$\sim p$	$\sim q$
$\exists F p$	p	$\exists X \exists F p$

Table 2: β-expansions

Remark: Comparing these tables to the expansion rules in [BPM], [CE], we see that the only rules which have changed are for formulae of the forms $\forall Fp$, $\exists Gp$. The example given in the appendix demonstrates the necessity for this change in the case of $\exists G$-formulae.

If none of these expansions are possible at n, then each element of F_n is either a proposition, a negated proposition, or of one of the forms $\forall Xp$, $\exists Xp$. In this case we call n a *state*, and we apply to it the X-expansion rule of [BPM]. That is, let

$$\forall Xp_1, \ldots, \forall Xp_a$$

be all the formulae preceded by $\forall X$ in F_n, and let

$$\exists Xq_1, \ldots, \exists Xq_b$$

be all the formulae preceded by $\exists X$ in F_n. Then create b sons $n_1, \ldots, n_b$ of n and put

$$F_{n_k} = \{p_1, \ldots, p_a, q_k\}$$

for each $k=1, \ldots, b$. (If $b = 0$, create one son n_0 of n and put

$$F_{n_0} = \{p_1, \ldots, p_a\}.)$$

Each successor of n under this expansion is called (as in [BPM]) a *pre-state*. The root n_0 is also called a pre-state.

The construction of T is terminated by using the same termination rules as in [BPM], i.e. not expanding

r	r_1	r_2	r_3
$p \wedge q$	p	q	
$\exists Gp$	p	$\exists X \exists G p$	$\forall X (true \vee \exists G p)$
$\sim(p \vee q)$	$\sim p$	$\sim q$	
$\sim(p \forall u q)$	$(\sim q) \exists u (\sim p)$		
$\sim(p \exists u q)$	$(\sim q) \forall u (\sim p)$		
$\sim(\forall F p)$	$\exists G \sim p$		
$\sim(\exists G p)$	$\forall F \sim p$		
$\sim(\forall X p)$	$\exists X \sim p$		
$\sim(\exists X p)$	$\forall X \sim p$		
$\forall G p$	p	$\forall X \forall G p$	

Table 1: α-expansions

and 'closing' nodes n for which F_n contains both a proposition and its negation, and, at an X-expansion of a node n, not creating new succeeding pre-states if their set of formulae is identical to the set of formulae of some ancestor pre-state m of n, in which case the corresponding outgoing edge from n points back to m. These termination rules ensure that the resulting graph T is finite, and that its size is at most exponential in the length of the initial formula p_0.

Having thus created T we proceed to mark some of its nodes using the following rules (the first four of which coincide with the rules (M1)-(M4) of [BPM], while the last two are special to the probabilistic case). Roughly speaking, a node is marked if its set of formulae cannot be satisfied by a model that can be obtained from "unwinding" the tableau. Such nodes will eventually be deleted from the tableau.

(M1) Mark every closed node (i.e. a node containing both a proposition and its negation.

(M2) If n is a node at which an α-expansion has been applied and its son n_1 has been marked then mark n.

(M3) If n is a node at which a β-expansion has been applied and both its sons n_1 and n_2 have been marked then mark n.

(M4) If n is a state and one of its succeeding pre-states has been marked then mark n.

(M5) Let $r = q \exists U p$ or $r = \forall Fp$, and let N_r be the set of all unmarked nodes n of T whose set of formulae F_n contains r. Assuming this set is nonempty, we apply the following 'ranking' algorithm to it:

(i) Initially, all nodes in N_r are unranked.

(ii) Let $n \in N_r$ be a node at which the β-expansion corresponding to r has been applied, and let n_1 be the son of n "inheriting" p. If n_1 is unmarked, give n_1 the rank 0, and give n the rank 1.

(iii) If $n \in N_r$ is a node at which an α-expansion or a β-expansion other than that in (ii) has been applied, and if one of the successors of n is ranked, then give n a rank which is 1 + the smallest rank of any son of n.

(iv) Finally, let $n \in N_r$ be a state (at which an X-expansion has been applied) and let n_1 be the son of n containing r if $r=p \exists U q$, or otherwise the son of n generated by the presence of the formula $\exists X \forall Fp$ in n. If n_1 has been ranked, then give n the rank of n_1.

After the completion of the ranking algorithm, all unranked nodes in N_r are marked.

Remark: Note that the marking rule for formulae of the form $\forall Fp$ is quite different from the corresponding rule in the nonprobabilistic case. In fact, a node n containing $\forall Fp$ will not be marked if there exists at least one path from n to a node containing p; in the nonprobabilistic case *all* paths from n must lead to a node containing p.

(M6) (This rule is not required at all in the nonprobabilistic case.) Let $r = \exists Gp$ be a formula appearing in the set F_n of some unmarked state n. We first introduce some notations: Without loss of generality, assume all marked nodes have been deleted from T. Let S denote the set of *states* in T, and let Π denote the set of *pre-states* in T. For every $s \in S$, let $X(s)$ be the set of all successor pre-states of s (obtained by the X-expansion rule); for each pre-state $\xi \in \Pi$, let $T(\xi)$ denote the set of all states in S which are reachable from ξ via paths consisting of α and β-expansions only. We will also use the inverse relations: $X^{-1}(\xi)$ denotes the set of all predecessor states of ξ (there may be more than one such state according to the rules for terminating the tableau construction), and $T^{-1}(s)$ is the (unique) pre-state preceding s. Essentially, all intermediate nodes of T which are neither states nor pre-states are ignored in the sequel.

Given $r = \exists Gp$ and $n \in S$ with $r \in F_n$ as above, let $S_r = \{s \in S : r \in F_s\}$, and let $Y \subset \Pi$ be the set of all pre-states ξ which are reachable from n along paths whose states all belong to S_r. We will obtain a decomposition of Y which is closely related to the decomposition of the state-space of a concurrent probabilistic program given in [HSP]. The purpose of this decomposition is to find *ergodic* sets E of states, all of which contain r, and for which there exists an "unwinding" of the tableau starting at any $s \in E$ and visiting from then on only states of E. Such an unwinding will enable us to show that r is satisfied in a model constructed from this unwinding and whose initial state is either in E or from which E can be reached via some finite path of states in N_r.

More precisely, define

$$I_0 = \{s \in T(Y) : r \notin F_s\}.$$

We will construct inductively a sequence of (disjoint) subsets $\{H_m\}_{m \geq 1}$ of Y, as follows. We begin by constructing a directed graph G, whose nodes are the pre-states of Y, and whose edges are given by the relation $\nu \equiv X \circ T$ restricted to Y (i.e. $\eta \in \nu(\xi)$ if there exists $s \in S_r$ such that $s \in T(\xi)$ and $\eta \in X(s)$; it is helpful to label each such edge by the corresponding state s); note that G may contain loops (i.e. edges of the form (ξ,ξ)) and multiple edges. Let H_1 be a terminal strongly connected component of G, including the degenerate case of a singleton $H_1 = \{\xi\}$, in which case it is not required that (ξ,ξ) be an edge of G. Thus, for each $\xi \in H_1$ and each $t \in T(\xi)$, either $X(t) \subset H_1$ or $t \in I_0$. Next, suppose that $H_1, \ldots, H_{m-1}$ have already been defined, and put $K_{m-1} = \bigcup_{i<m} H_i$. We first update G by erasing all nodes $\xi \in H_{m-1}$, together with all edges (η,η') for which there exists $s \in T(\eta)$ such that both ξ and η' belong to $X(s)$ (thus, besides edges (η,ξ) we also erase edges (η,η') with the same label s as (η,ξ)). H_m is then defined to be a terminal strongly connected component of the (updated) graph G (including the degenerate case of a singleton, as above). Thus, H_m has the following property: For each $\xi \in H_m$ and each $t \in T(\xi)$, either

(1) $t \in I_0$; or

(2) $X(t) \cap K_{m-1} \neq \emptyset$; or

(3) $X(t) \subset H_m$. (Note that this holds for $m = 1$ too.)

We continue with this process until G becomes empty.

Having obtained this decomposition, we next define, for each $m \geq 1$

$$I_m = \{t \in S : T^{-1}(t) \in H_m \text{ and } X(t) \subset H_m\}.$$

$$L_m = \{t \in S : T^{-1}(t) \in H_m, t \notin I_0 \text{ and } X(t) \cap H_m\}$$

It is easy to establish the following properties:

(a) $I_m = \emptyset$ iff H_m is a non-strongly connected singleton $\{\xi\}$; in this case each $t \in T(\xi)$ is either in I_0 or satisfies $X(t) \cap K_{m-1} \neq \emptyset$.

(b) For each $s \in I_m$ and each $\xi \in X(s)$ we have $T(\xi) \cap I_m \neq \emptyset$.

(c) For each $s,t \in I_m$, $t \neq s$, there exists a chain of states in I_m, $s=s_0,s_1, \ldots ,s_j=t$; such that $s_{i+1} \in N(s_i)$, for $i=0, \ldots ,n-1$, where $N = T \circ X$.

Suppose that $I_m \neq \emptyset$ for some m. Intuitively, this means that, starting at some $s \in I_m$, one can "unwind" the tableau into an infinite tree which consists only of states in I_m, by choosing at each pre-state $\xi \in H_m$ a state $t \in T(\xi) \bigcap I_m$, and by noting that all successor pre-states of t are contained in H_m. Since the formula $\exists G p$ is contained in F_s, we could potentially use such an unwinding of T as a model for the satisfiability of $\exists G p$. This, however, depends on our ability to satisfy other formulae of F_s by that same unwinding. As will be seen below, it suffices to require from I_m that its unwinding can satisfy every formula of the form $\forall F q$ which appears at some of its states.

Definition: A set E of states in S is called an **ergodic set** if it satisfies properties (b) and (c) stated above for I_m, and moreover for each formula of the form $\forall F q$ which appears in F_s for some $s \in E$, there exists $t \in E$ such that $q \in F_t$.

Consequently, we distinguish between three subcases:

(i) $I_m = \emptyset$.

(ii) I_m is ergodic.

(iii) I_m is not ergodic. That is, there exists $s \in I_m$ and a formula $(\forall F q) \in F_s$ such that $q \notin F_t$ for all $t \in I_m$. (It is easily seen, by the properties of expansions involving $\forall F$, that in this case the formula $\forall F q$ belongs to F_t for every $t \in I_m$.)

We are now in a position to state the marking rule (M6) for $r = \exists G p$ and for a state $n \in S$ containing r:

Obtain the above decomposition of the set Y, and check for each $m \geq 1$ for which I_m is nonempty whether this set is ergodic. If no such ergodic set exists (i.e. for each m either case (i) or case (iii) holds), then mark n. Otherwise n remains unmarked.

The marking process proceeds in phases; in each such phase we either apply one of the rules (M1)-(M4) to a single node of T, or apply rule (M5) to a formula $\forall F p$ or $p \exists U q$ at some node of T, which may cause several nodes of T to be marked simultaneously, or apply rule (M6) to a formula $\exists G p$ and a node containing it, which again can result in marking more than one node. This marking process terminates when no new nodes can be marked. An example illustrating the tableau construction and marking rules is given in an appendix below. The main result of this paper is the following

Theorem 4.1: p_0 is satisfiable if and only if the root n_0 of T has not been marked. Moreover, if the root has been marked then $\sim p_0$ is provable in the axiomatic system of section 3 (i.e. this system is complete). In this latter case the proof of $\sim p_0$ can be obtained mechanically off the tableau T.

The proof of this theorem is fairly involved, and is therefore only very briefly sketched in this abstract, with most of the technical detail omitted.

We first show that if n_0 has not been marked, then we can construct a model of PTL_b for p_0 from the unmarked nodes of T. This is achieved by first constructing from the unmarked nodes of T a *Hintikka structure* (defined below), and then transforming this structure into a model for p_0.

Definition: A *Hintikka structure H* for a formula p_0 of PTL_b is an infinite tree with a root s_0 such that the number of sons of any node in H is bounded, and such that with each node $s \in H$ there is associated a set F_s of formulae of PTL_b. Given such a tree H, we can associate with each edge (m,n) of H a *transition probability* equal to $1/d$, where d is the out-going degree of m (note that these probabilities are bounded away from 0). This probability assignment allows us to regard H as a stochastic process, and induces, for each node $s \in H$, a probability measure $\mu_{s,H}$ on the set Ω_s of all infinite paths in H starting at s, in the standard manner as in Section 2. In addition, H must have the following properties ($p \in F_s$ is abbreviated as $p \in s$):

(H0) $p_0 \in s_0$.

(H1) $\sim p \in s$ implies $p \notin s$ (i.e. H is consistent).

(H2) Let r be a formula to which an α-expansion is applicable (see Table 1); then $r \in s$ implies $r_j \in s$ for all corresponding subconjuncts r_j of r (appearing in the other columns of the table).

(H3) Let r be a formula to which a β-expansion is applicable (see Table 2); then $r \in s$ implies $r_1 \in s$ or $r_2 \in s$ (where r_1,r_2 are the two corresponding disjuncts appearing in the other columns of the table).

(H4a) If $\forall X p \in s$ then $p \in t$ for all succeeding nodes t of s in H.

(H4b) If $\exists X p \in s$ then $p \in t$ for at least one succeeding node t of s in H.

(H4c) If $p \exists U q \in s$ then there exists a path from s all of whose nodes contain p, and its last node also contains q.

(H4d) If $p \forall U q \in s$ then every simple path starting at s either contains p in all its nodes, or contains p at all its initial nodes (possibly none) before reaching a node which contains q.

(H4e) If $\forall F p \in s$ then there exists a stopping time N, defined on Ω_s(the subtree of H rooted at s), which is $\mu_{s,H}$-almost-surely finite, and for which $p \in \omega_{N(\omega)}$, for each $\omega \in \Omega_s$ with $N(\omega) < \infty$.

(H4f) If $\exists G p \in s$ then

$$\mu_{s,H}\{\omega \in \Omega_s : p \in \omega_n \text{ for all } n \geq 1\} > 0.$$

Lemma 4.2: If p_0 has a Hintikka structure, then it has a model, i.e. it is satisfiable.

Proof: Omitted.

It therefore remains to construct a Hintikka structure H for p_0 from the unmarked nodes of T. For this, we use the following construction, in which we assume, for simplicity, that all nodes of T are unmarked. The following observations, which have already appeared implicitly in the marking rule (M6), will be useful in motivating and explaining the construction of the required Hintikka structure. As before, we let S (resp. Π) denote the set of all (unmarked) states (resp. pre-states) of T. The nodes of the Hintikka structure H we are about to construct from T will be states in S. The number of sons of a node $s \in H$ is the same as the number of pre-states in $X(s)$. For each such $\xi \in X(s)$ there will correspond a son of s in H which will be an element of $T(\xi)$. The decisions as to which state in $T(\xi)$ to choose as the corresponding son of s can be thought of as being taken by some "scheduler", and we will refer to them as a *schedule* of T. This notation is very similar to the modelling of the execution of a concurrent probabilistic program, as described e.g. in [HSP]. In this analogy, the "program states" are our pre-states Π; at each such $\xi \in \Pi$, the schedule assigns a process to execute the next program state, which, in our case, corresponds to choosing a state $t \in T(\xi)$, and then "execute" the X-transitions from t to new pre-states (i.e. new program states). Thus "program execution" corresponds to the construction of H, in which we just record the states in S chosen by the scheduler.

The preceding remarks imply that to construct H it suffices to define the corresponding schedule σ. σ is a function defined on the set of all *finite execution histories*, each such history being a sequence of the form

$$h_n = (\xi_0, s_1, \xi_1, s_2, \ldots, s_n, \xi_n)$$

with ξ_0 the root of T and where for each $i=1, \ldots, n$ we have $s_i \in T(\xi_{i-1})$ and $\xi_i \in X(s_i)$. (Thus, for convenience, we label each node of H also by the pre-state ξ of its corresponding state s. Each such h_n corresponds to a path ω of length n in H in which the schedule's decisions at the first $n-1$ nodes are already recorded; $\sigma(h_n)$ is to be the state s_{n+1} in $T(\xi_n)$ that the schedule will choose at the terminal node of ω.)

Before defining σ, we first modify T slightly to eliminate any partial overlapping between ergodic sets. For this, suppose that E_1 and E_2 are two distinct ergodic sets (for the same, or for different formulae) whose intersection $E = E_1 \cap E_2$ is nonempty. We then duplicate each $s \in E$ into two copies $(s,1)$ and $(s,2)$ such that for $j=1,2$ we have

$$F_{(s,j)} = F_s; \quad X((s,j)) = X(s); \quad T^{-1}((s,j)) = T^{-1}(s).$$

Having thus split each $s \in E$, we define

$$E_1' = (E_1 - E_2) \cup \{(s,1) : s \in E\},$$

and

$$E_2' = (E_2 - E_1) \cup \{(s,2) : s \in E\}.$$

Since the internal structure of E_j' is isomorphic to the structure of E_j for $j=1,2$, it follows that both these new sets are ergodic. Furthermore, if we apply the marking procedure to the new tableau obtained by this splitting, then no new nodes will be marked because the duplication of states in the above manner cannot cause any of the marking rules (M1)-(M6) to become applicable if it were not applicable before.

Repeating the splitting procedure just described as needed, we obtain an equivalent but larger tableau T' for which all ergodic sets are pairwise disjoint. Without loss of generality, we will assume that T itself already has this property. Let $E_1, \ldots, E_d$ be the ergodic sets of T. Then each set $s \in S$ either belongs to a unique ergodic set E_e, or else is "transient", i.e. belongs to $E_0 = \left(\bigcup_{e=1}^{d} E_e \right)^c$.

For every $s \in S$ and $\xi \in X(s)$, we define

$$V(s,\xi) = \begin{cases} T(\xi) \cap E_e, & \text{if } s \in E_e,\ e>0 \\ T(\xi), & \text{if } s \in E_0 \end{cases}.$$

Note that $V(s,\xi)$ is always nonempty.

For each such s and ξ let $(t_\xi^1, \ldots, t_\xi^k)$ be a fixed enumeration of the elements of $T(\xi)$, and let $(v_{(s,\xi)}^1, \ldots, v_{(s,\xi)}^l)$ be a fixed enumeration of the elements of $V(s,\xi)$ (note that $k=k(\xi)$ and $l=l(s,\xi)$). Let

$$h_n = (\xi_0, s_1, \xi_1, s_2, \ldots, s_n, \xi_n)$$

be a finite history in H; we will define $\sigma(h_n) = s_{n+1}$ as follows. Let r be the number of occurrences of $s = s_n$ in h_n (up to and including n), i.e. $r = |\{i : 1 \le i \le n,\ s_i = s_n\}|$. Two possible cases can arise:

(a) All the following three conditions hold:

- (*i*) $r = \nu^2$ for some $\nu \ge 3$.
- (*ii*) $|X(s)| > 1$.
- (*iii*) $\xi_{m_1} = \xi_{m_2} = \cdots = \xi_{m_\nu}$, where $n > m_1 > m_2 > \cdots > m_\nu$ are the places in h_n of the last ν occurrences of s (before the n-th place).

(b) At least one of these conditions does not hold.

If case (a) occurs, let $j \equiv \nu \pmod{k}$ and define $\sigma(h_n) = t_{\xi_n}^j$.

If case (b) occurs, let $j \equiv (r - \lfloor \sqrt{r} \rfloor) \pmod{l}$ and define $\sigma(h_n) = v_{(s_n,\xi_n)}^j$.

Let us call a visit at s for which r is a perfect square ≥ 9 a *square* visit. Thus case (a) occurs only at pre-states ξ following a square visit (of order ν^2) at a state s which has more than one succeeding pre-state, and for which s is followed in h_n by the same pre-state ξ at each of the last ν visits at s. When this case applies, the schedule iterates through $T(\xi)$ in a round robin fashion (stepping through the elements of $T(\xi)$ once per each such special square visit). Similarly, case (b) occurs when the visit at the last state s in h_n is either non-square, or is square but not all last $\sqrt{r}$ visits at s have been followed by the same succeeding pre-state. In these cases the schedule iterates through $V(s,\xi)$ in a round robin fashion, in which the square visits at s are not counted.

We next show that the unwinding of the tableau T by the schedule σ just defined yields a Hintikka structure H for p_0. The proof that H satisfies conditions (H4c),

(H4e) and (H4f) is somewhat involved and technical. There are some difficulties in showing the existence of a path (or, in the case of (H4f), a set of paths having positive measure) passing only through nodes containing the corresponding subformula $r = q \exists U p$, $\forall Fp$ or $\exists Gp$ and (in the cases of (H4c) and (H4e)) ending at a node containing p; these difficulties arise because as such a path is being constructed, it can enter various ergodic sets, some of which may be irrelevant to the "fulfillment" of r, and so must be exited in order for r to be fulfilled. The schedule σ has been defined in a way which ensures that these sets are properly exited from, and that the required path or set of paths does exist. These technical details are omitted in this version, and we just state the final conclusion.

Theorem 4.3: H is a Hintikka structure for p_0. Thus p_0 is satisfiable.

We have thus shown that if the root of T has not been marked then p is satisfiable. The converse statement is proven in the following section.

5. Completeness

In this section we prove the second part of Theorem 4.5, namely that if the root n_0 of the tableau T constructed for a formula p_0 of PTL_b is marked by the procedure described in the preceding section, then $\sim p_0$ is provable from the axioms of PTL_b given in section 3. This, together with the proof of the first part of Theorem 4.5, will establish the completeness of these axioms.

As in [BPM] we define, for each node $n \in T$, the *associated formula* af_n of n to be $\vee\{\sim p : p \in F_n\}$; note that $af_{n_0} = \sim p_0$. The proof proceeds by showing, using induction on the phases of the marking procedure, that if n is a marked node, then af_n is provable. Since we assume that n_0 is marked, it follows that $\sim p_0$ is provable.

The basis for our induction are phases which mark nodes n using the rule (M1). In these cases we obtain, using dilution as in Lemma 5.1 of [BPM], that $\vdash af_n$. Similarly, for phases which mark nodes n using one of the rules (M2)-(M4), we can show, as in Lemma 5.2 of [BPM], that $\vdash af_n$. For the rules (M2) and (M3) (corresponding respectively to α and β expansions) this follows from simple propositional reasoning and from the fact that each of the expansions listed in Tables 1 and 2 of Section 4 is a theorem of PTL_b. For the rule (M4) (corresponding to X-expansions) the proof proceeds exactly as in [BPM], using rule (R1') and theorem (T1).

Next consider a marking phase which has applied rule (M5) to a formula $r = p \exists U q$. Let t be a state in T which has been marked because it has not been ranked. In what follows we will ignore the marked portion of T (before the current application of (M5)), and assume that each node we refer to is presently unmarked. Let us introduce the following terminology: For each state $u \in S$ such that $r \in F_u$, denote the r-son of u (i.e. that son inheriting r) by η_u, and put $R(u) = T(\eta_u)$ (i.e. the *states* following η_u). Also put $R^* = R^*(t) = \bigcup_{m=0}^{\infty} R^m(t)$. Note that each state in R^* has not been ranked, for otherwise the ranking algorithm of (M5) would have ranked t too.

For each $u \in R^*$ define $V(u)$ to be the set of (presently unmarked) nodes at which r is expanded, which are reachable from η_u by α and β expansions only (i.e. before a state in $R(u)$). Note that the essential son v_1 of any $v \in V(u)$ has already been marked, for otherwise v, and consequently also u and t, would be ranked by (M5). For each $v \in V(u)$, define $Q(v)$ to be the set of unmarked states reachable from v (or from the nonessential son v_2 of v) by α and β expansions only. Note that $Q \circ V = R$ for all $u \in R^*$. Also denote, for each state $u \in R^*$,

$$Y_u = \{k : \forall Xk \in F_u\}$$

$$W_u = \bigwedge_{k \in Y_u} k \equiv \wedge Y_u \text{ (for short)}$$

and

$$W^t = \bigvee_{u \in R^*} W_u .$$

These formulae have the following properties, generalizing Lemmas 5.3 and 5.4 of [BPM]:

Lemma 5.1: $\vdash W^t \supset \sim p \vee \sim q$.

Proof: Omitted.

Lemma 5.2: $\vdash p \wedge W^t \supset \forall X W^t$.

Proof: Omitted.

We can now show that $\vdash af_t$, as follows. Note that $F_{\eta_t} = \{r\} \cup Y_t$. Thus $af_{\eta_t} = \sim r \vee \sim W_t$. We have, by Lemmas 5.1 and 5.2,

$$\vdash W^t \supset \sim p \vee \sim q ,$$

$$\vdash W^t \supset \sim p \vee (\forall X W^t) .$$

Hence

$$\vdash W^t \supset \sim p \vee \left(\sim q \wedge \forall X W^t\right) ,$$

or, using (R2),

$$\vdash W^t \supset (\sim q) \forall U (\sim p) .$$

Hence we also have

$$\vdash W_t \supset \sim (p \exists U q)$$

or

$$\vdash \sim W_t \vee \sim r$$

that is, $\vdash af_{\eta_t}$. Hence af_t too, as follows from the inductive step of our proof corresponding to the marking rule (M4).

Next consider a marking phase which has applied rule (M5) to a formula $r = \forall Fp$, and let t be a state which has been marked by this phase. In a similar manner to what we did above for formulae involving $\exists U$, we denote by $R^* = R^*(t)$ the set of all (presently unmarked) states reachable from t by choosing at each state u the r-son of u (also to be denoted as η_u). Again, since t has been marked at this phase, no state in R^* has been ranked, so that it must contain both $\forall Xr$ and $\exists Xr$. In other words, at each node v where r has been expanded, the essential $\forall F$-son of v must have been previously marked. We will also use the notations $R(u)$, $V(u)$, $Q(v)$ as above, each of which is defined in an obviously modified manner. Also put

$$Y_u = \{k : \forall Xk \in F_u \text{ and } k \neq r\}$$

$$W_u = \bigwedge_{k \in Y_u} k = \wedge Y_u$$

$$W^t = \bigvee_{u \in R^*} W_u$$

These formulae have the following properties:

Lemma 5.3: $\vdash W^t \supset \sim p$.

Proof: Omitted.

Lemma 5.4: $\vdash W^t \supset \forall X W^t$.

Proof: Omitted.

Lemmas 5.3 and 5.4 together imply

$$\vdash W^t \supset (\sim p) \wedge \forall X W^t$$

which, using (R2), imply that

$$\vdash W^t \supset (\sim p) \forall U \text{ false}$$

which in turn implies (using the contrapositive form of (T8) and (R0))

$$\vdash W^t \supset \exists G \sim p$$

that is

$$\vdash \sim (W_t \wedge \forall F p) .$$

In other words, we have shown that $\vdash af_{\eta_t}$, from which, using the argument corresponding to the marking rule (M4), it follows also that $\vdash af_t$, which is what was to be shown.

Finally, we need to consider nodes at which the marking rule (M6) has been applied to some formula $r = \exists G p$. This portion of the proof is the most complex, and is special to PTL_b, in the sense that all the other related logics (*UB*, *CTL*, PTL_f) have no similar marking rule for $\exists G p$. In this version we will only give a very brief sketch of the proof.

Let $t \in S$ be a state that has been marked by (M6) for the formula r. Let $\Gamma(t)$ denote the set of all pre-states reachable from t along paths each of whose *states* contain r. Let $R^*(t) = \{t\} \cup T[\Gamma(t)]$. (Recall the notations introduced when (M6) was defined at Section 4; for a state s, $X(s)$ is the set of its sons (pre-states); for a pre-state ξ, $T(\xi)$ is the set of states reachable from ξ by α and β expansions only.)

Applying rule (M6) for t and r, we decompose $\Gamma(t)$ into finitely many disjoint sets $H_1 \cup H_2 \cdots \cup H_m$, and obtain a similar decomposition of $R(t)$ into $I_0 \cup I_1 \cdots \cup I_m \cup L_1 \cdots \cup L_m$ where I_0 is the set of all states in $R(t)$ which do not contain r, where each of the sets $I_1, \ldots, I_m$ is either empty or a communicating but nonergodic set of states, and where $L_1, \ldots, L_m$ are "transition sets" of states satisfying

$$s \in L_j \Leftrightarrow T^{-1}(s) \in H_j \text{ and } X(s) \cap K_{j-1} \neq \emptyset$$

(recall that $L_1 = \emptyset$).

As already noted, since t has been marked by (M6), we must have, for each $j=1, \ldots, m$ either

(I) $I_j = \emptyset$; or

(II) $I_j \neq \emptyset$ and there exists some formula q such that for each $s \in I_j$, $\forall F q \in F_s$ but $q \notin F_s$.

Before investigating both these cases, we begin with a few general observations and notations. For each $\xi \in \Gamma(t)$ we denote

$$Z_\xi = \wedge F_\xi$$

and for each $s \in R(t)$ we denote

$$Q_s = \wedge F_s .$$

Let $X(s) = \{\eta_1, \ldots, \eta_k\}$. This means that Q_s is a conjunction involving, among others, formulae of the form

$$\exists X \beta_{\eta_1}, \ldots, \exists X \beta_{\eta_k} ,$$

and also formulae of the form $\forall X k$, where $k \in Y_s$, in the sense that for each $j=1, \ldots, k$ we have

$$F_{\eta_j} = Y_s \cup \{\beta_{\eta_j}\} .$$

Note also that we must have one j for which $\beta_{\eta_j} = \exists G p$; we may assume $j = 1$. As usual we denote $W_s = \wedge Y_s$, so that we can write

$$\vdash Q_s \supset \forall X W_s \wedge \exists X \beta_{\eta_1} \wedge \cdots \wedge \exists X \beta_{\eta_k} \qquad \text{(G1)}$$

which leads to

$$\vdash Q_s \supset \forall X \Big[(W_s \wedge \beta_{\eta_1}) \vee (W_s \wedge \beta_{\eta_2}) \vee \cdots \vee (W_s \wedge \beta_{\eta_k}) \qquad \text{(G2)}$$
$$\vee (W_s \wedge \sim \beta_{\eta_1} \wedge \cdots \wedge \sim \beta_{\eta_k}) \Big]$$
$$\wedge \exists X (W_s \wedge \beta_{\eta_1}) \wedge \cdots \wedge \exists X (W_s \wedge \beta_{\eta_k})$$

But $\sim \beta_{\eta_1} \equiv \sim \exists G p$, so that we can write (G2) as

$$\vdash Q_s \supset \forall X (Z_{\eta_1} \vee Z_{\eta_2} \cdots \vee Z_{\eta_k} \vee \sim \exists G p) \wedge \qquad \text{(G3)}$$
$$\exists X Z_{\eta_1} \cdots \wedge \exists X Z_{\eta_k} .$$

Let us define, for each $j=1, \ldots, m$,

$$A_j = \bigvee_{\xi \in H_j} Z_\xi \quad ; \quad B_j = \bigvee_{i \leq j} A_i .$$

Let us fix some $j=1, \ldots, m$, and consider both cases (I) and (II) listed above:

(I) $I_j = \emptyset$. In this case H_j must contain a single element ξ, and $T(\xi) \subset I_0 \cup L_j$.

Lemma 5.5: If I_j satisfies condition (I) then

$$\vdash A_j = Z_\xi \supset \bigvee_{s \in T(\xi) \cap L_j} Q_s \vee \qquad \text{(G4)}$$
$$\Big[\bigvee_{s \in T(\xi) \cap I_0} (Q_s \wedge \sim \exists G p) \Big]$$

Proof: Omitted.

From these formula we can deduce that for I_j's satisfying (I) we have

$$\vdash A_j \supset \Big[\forall X (B_m \vee \sim \exists G p) \wedge \exists X B_{j-1} \Big] \vee \sim \exists G p . \qquad \text{(G5)}$$

(II) $I_j \neq \emptyset$ and there exists q for which $\forall F q \in F_s$ and $q \notin F_s$ for each $s \in I_j$.

Let $\xi \in H_j$ be any pre-state. As in case (I) we claim

Lemma 5.6: If I_j satisfies condition (II) then

$$\vdash Z_\xi \supset \bigvee_{s \in T(\xi)\cap I_j}(Q_s \wedge \sim q) \vee \Big(\bigvee_{s \in T(\xi)\cap L_j} Q_s\Big) \vee \Big(\bigvee_{s \in T(\xi)\cap I_0}(Q_s \wedge \sim \exists G p)\Big) \qquad \text{(G6)}$$

Proof: Omitted.

These results, plus a few additional arguments imply that for I_j's satisfying (II) we have

$$\vdash A_j \supset \forall F\left(\sim\exists G p \vee \left[\forall X(B_m \vee \sim\exists G p) \wedge \exists X B_{j-1}\right]\right). \qquad \text{(G7)}$$

and since (G5) is a special case of (G7) by (A4), we conclude that (G7) holds for each $j=1, \ldots, m$. From this we can show that

$$\vdash B_j \supset \forall F\left(\sim\exists G p \vee \left[\forall X(B_m \vee \sim\exists G p) \wedge \exists X B_{j-1}\right]\right) \qquad \text{(G8)}$$

for each $1<j\leq m$. For $j=1$, the term in square brackets disappears, because L_1 is empty; in this case (G8) reduces to

$$\vdash B_1 \supset \forall F(\sim p) \qquad \text{(G9)}$$

Lemma 5.7:

$$\vdash B_m \supset \forall F(\sim p) \qquad \text{(G10)}$$

Proof: Omitted in this version; nevertheless we remark that it is here where rule (R4) is needed to establish completeness of our axiomatic system.

Having established (G10), choose next any pre-state $\xi \in \Gamma(t)$ such that $\exists G p \in F_\xi$ (e.g. for each $s \in = R(t)-I_0$, the son η_s of s corresponding to $\exists X \exists G p \in F_s$ is such a pre-state). For each such ξ we have

$$\vdash Z_\xi \supset B_m \supset \forall F(\sim p)$$

and also

$$\vdash Z_\xi \supset \exists G p$$

both of which formulae imply that

$$\vdash \sim Z_\xi$$

so that, by dilution,

$$\vdash af_\xi$$

Thus, by the portion of the completeness proof corresponding to the marking rule (M4), it follows that $\vdash af_s$ for each $s \in R'(t)$, and in particular

$$\vdash af_t$$

Q.E.D.

This concludes our proof of completeness, for we have shown that the associated formula of each marked node is provable, and in particular $af_{n_0} = \sim p_0$ is provable, which is what we wanted to show. Q.E.D.

6. Discussion

Several additional consequences of our results deserve comment: First, the arguments of Section 4 actually imply that PTL_b has the following

finite pre-model property: If a formula p of PTL_b is satisfiable, then there exists a finite set of states I (actually pre-states of the associated tableau), and for each $s \in I$ there is a finite collection $K(s)$ of probability distributions over I such that a model for p can be obtained by some inductive "strategy" of choosing a distribution out of $K(s)$ at each state s which in turn adds all states in the support of that distribution as successors of s in the model.

The interest in the finite pre-model property stems from the intended application of PTL_b to argue about concurrent probabilistic programs. If p is a formula which asserts some property of a concurrent probabilistic program, and which we wish to prove in PTL_b, then either p is indeed provable, or else, by the finite pre-model property applied to $\sim p$, we can effectively construct a *finite−state* concurrent probabilistic program and produce a certain scheduling of its processes for which execution p does not hold. Thus PTL_b, although interpreted over a larger collection of models, does serve as the proper tool to argue about finite-state concurrent probabilistic programs. This observation also implies the following

Corollary: It is impossible to express in PTL_b a property of concurrent probabilistic programs which is true for some such programs having *infinitely* many states, but is false for all finite-state programs.

However, the above model for a satisfiable formula p need not itself be finite (i.e. a model of PTL_f), as can be seen from the example given in the appendix below. In fact, the formula $\forall G\exists F p \wedge \exists G \sim p$ has no finite model, as can be easily checked. Nevertheless, it is possible to give sufficient and purely syntactic conditions for the existence of a finite model for a satisfiable formula of PTL_b. In particular we show that formulae expressing termination of finite state concurrent probabilistic programs satisfy these conditions, and hence can be tested for satisfiability in PTL_f (for which a simpler decision procedure is available). More details are given in the complete version of the paper.

References

[BPM] M. Ben-Ari, A. Pnueli and Z. Manna, *The Temporal Logic of Branching Time*, Tech. Rept., Dept. of Applied Math., Weizmann Institute of Science, 1982.

[CE] E.M. Clarke and E.A. Emerson, Design and Synthesis of Synchronization Protocols using Branching Time Temporal Logic, *Proc. Workshop on Logics of Programs*, D. Kozen (Ed.), Springer Verlag 1982.

[FH] Y. Feldman and D. Harel, A Probabilistic Dynamic Logic, *Proc. 14th Symp. Theory of Computing*, 1982, pp. 181-195.

[Fe] Y. Feldman, A Decidable Propositional Probabilistic Dynamic Logic, *Proc. 15th Symp. Theory of Computing*, 1983, pp. 298-309.

[HSP] S. Hart, M. Sharir and A. Pnueli, Termination of Concurrent Probabilistic Programs, *ACM Trans. Prog. Lang. and Systems*. 5(1983), pp. 356-380.

[HS] S. Hart and M. Sharir, *Concurrent Probabilistic Programs, or: How to Schedule if You Must,* Tech. Rept., School of Math. Sciences, Tel Aviv University, May 1982.

[Ko] D. Kozen, A Probabilistic PDL, *Proc. 15th Symp. Theory of Computing,* 1983, pp. 291-297.

[KL] S. Krauss and D. Lehmann, Decision Procedures for Time and Chance, *Proc. 24th Symp. Foundations of Computer Science* 1983, pp. 202-209.

[LS] D. Lehmann and S. Shelah, Reasoning with Time and Chance, *Information and Control* 53(1983) pp. 165-198.

[Pn] A. Pnueli, On the Extremely Fair Treatment of Probabilistic Algorithms, *Proc. 15th Symp. Theory of Computing,* 1983, pp. 278-290.

[SPH] M. Sharir, A. Pnueli and S. Hart, The Verification of Probabilistic Programs, *Siam J. Computing* (to appear).

Appendix: An Example.

Let $r = \forall G \exists F p \wedge \exists G \sim p$ be a formula in PTL_b (for simplicity we use $\forall G$, $\exists F$ explicitly, instead of their implicit representation by $\forall U$, $\exists U$.) The construction of the tableau T for r is given in Fig. 1. To make T more compact and readable, we combine several successive α expansions into a single expansion. Pre-states are given numerical labels enclosed in circles, while states are given alphabetical labels enclosed in boxes. Some intermediate nodes are also labeled, to indicate that the associated set of formulae is identical to that of a previously constructed node, so that further elaboration of T from that node on can be omitted from the figure. α and β expansions are denoted by dashed edges, while X expansions are denoted by solid edges.

The tableau T just constructed is given in condensed form in Fig. 2, where only pre-states and states are shown, and where the expansion of a pre-state is shown only once.

Since the subformula $\exists F p$ is "fulfilled" at state c, and since c is reachable from any other node of T, application of the marking rule (M5) results in no marking. Application of the rule (M6) to the subformula $q = \exists G \sim p$ proceeds as follows: The set N_q of relevant nodes is $\{1,2,a,a',b,c\}$. The decomposition of this set as specified in (M6) is:

$$I_0 = \{b,c\}; \quad H_1 = \{1,2\}; \quad I_1 = \{a,a'\}$$

Since no subformulae of the form $\forall F$ appear in r, the set I_1 is ergodic, and so no node needs to be marked by this

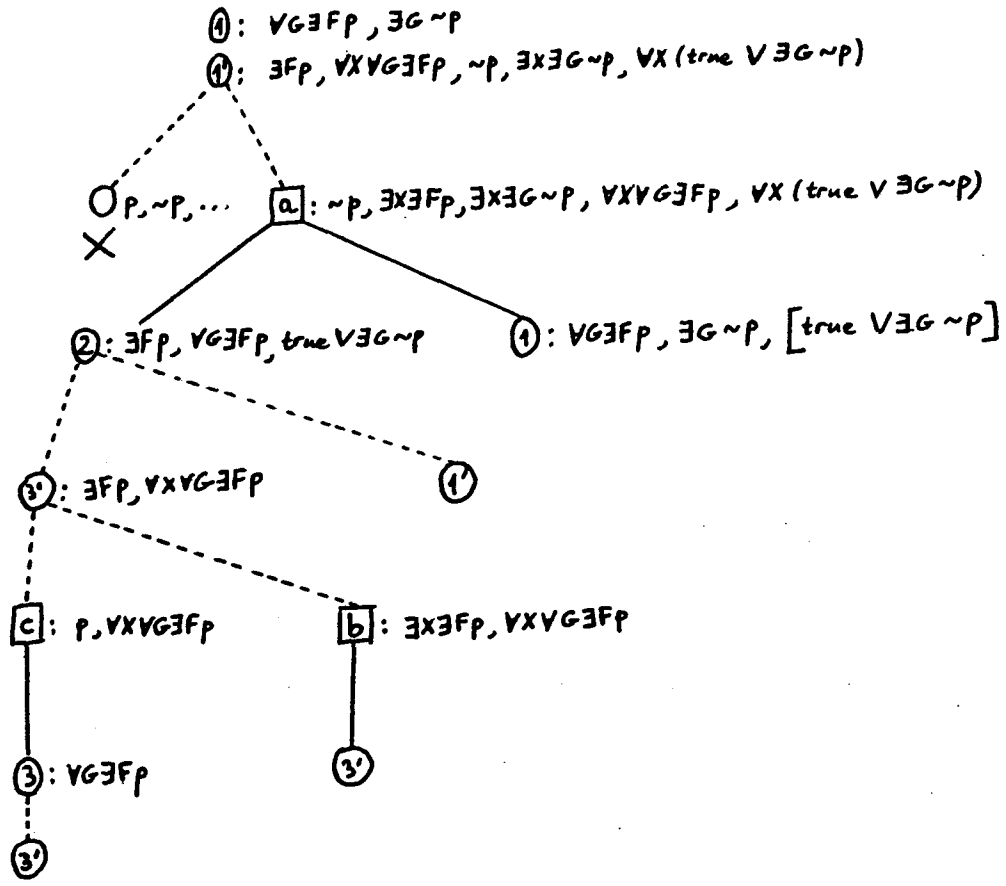

Fig. 1. The tableau of r

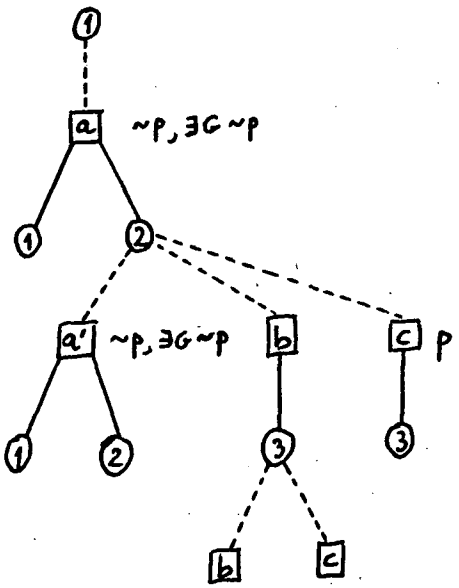

Fig. 2. A condensed form of the tableau.

rule. Hence r is satisfiable. To obtain a model for r from T we can use the following "scheduling" strategy for the unwinding of T:

At the pre-state 1 schedule the (only possible) state a.

At the pre-state 2 schedule the state a' until the first time in which the number of visits at the pre-state 2 equals the number of visits at the pre-state 1, in which case schedule the state c.

At the pre-state 3 always schedule the state c.

The resulting model is shown in Fig. 3. It essentially coincides with the behaviour of a random walk on the nonnegative integers with absorption at 0. This model will satisfy r provided that we assign a probability $> 1/2$ to the edges from a to 1 and from a' to 1. The reader is invited to check that if one expands subformulae of the form $\exists Gq$ as $q \wedge \exists X \exists Gq$ (as done in the nonprobabilistic case [BPM]), then the modified tableau for r would be such that no model for r could be obtained by its unwinding.

Remark: The schedule just introduced does not coincide with, and in fact is much simpler than the general schedule given in Section 4. The "price" that we pay for this simplicity is the need to assign probabilities within specific ranges to the transitions of this schedule, instead of the uniform assignment rule used in the schedule of Section 4.

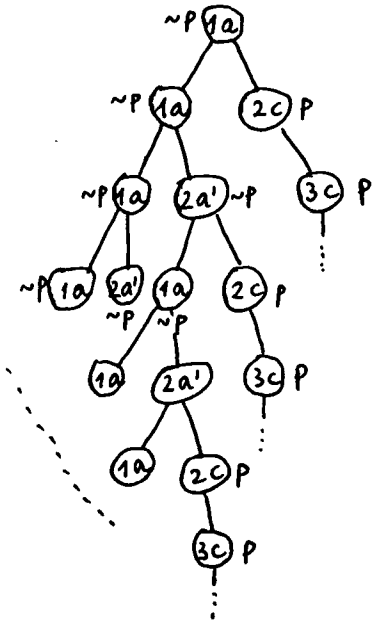

Fig. 3. Unwinding T into a model for r.