It Usually Works: The Temporal Logic of Stochastic Systems

Adnan Aziz Vigyan Singhal Felice Balarin
Robert K. Brayton Alberto L. Sangiovanni-Vincentelli *
Email: {adnan,vigyan,felice,brayton,alberto}@cs.berkeley.edu
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720, USA

Abstract. In this paper the branching time logic pCTL* is defined. pCTL* expresses *quantitative* bounds on the probabilities of correct behavior; it can be interpreted over discrete Markov processes. A bisimulation relation is defined on finite Markov processes, and shown to be sound and complete with respect to pCTL*. We extend the universe of models to generalized Markov processes in order to support notions of refinement, abstraction, and parametrization. Model checking pCTL* over generalized Markov processes is shown to be elementary by a reduction to RCF. We conclude by describing practical and theoretical avenues for further work.

1 Introduction

The study of formal methods to specify and prove properties of finite state systems has been the subject of intense research. Various methodologies have been proposed; some of the most fruitful, in both theory and practise, have been based on *temporal logic* [10]. Properties are expressed using formulae which are built out of operators on the computation paths.

The goal of this paper is to take verification techniques developed for temporal logic, and apply them to *stochastic systems*, i.e. systems in which there is a certain probability associated with events. For a large class of such systems, conventional logics interpreted over finite Kripke structures are insufficient.

Consider for instance a network which can support at most 10 user requests at a time and drops any additional requests. If 50 users interface to the network, and they can freely generate requests, then in the usual formulations of temporal logic and system models (and even the probabilistic temporal logics referenced below) it will be false that requests are always acknowledged, since there exists a finite probability of failing to service a request. However, a more accurate analysis should take into account the fact that the user requests have a statistical distribution resulting in a low likelihood of a large number of simultaneous requests, and that it is acceptable to drop requests provided the probability of this is small.

* Supported by SRC Grant 94-DC-008 and NSF/DARPA Grant MIP-8719546

In this paper several contributions are made towards formal reasoning about stochastic systems. We define pCTL*, a probabilistic variant of Computational Tree Logic [10]. The logic expresses *quantitative* stochastic properties of systems, which are themselves modelled as discrete Markov processes; furthermore, it exhibits an elementary model checking procedure. Discrete Markov processes exhibit a natural notion of *bisimulation*; this is shown to be sound and complete with respect to pCTL*. We also extend the universe of models to include *generalized discrete Markov processes* which can be used for modelling systems where the transition probabilities are not completely specified; these systems allow notions of abstraction and refinement. Using characterizations of discrete Markov processes developed in [15] and results on the decidability of real closed fields [5] we derive an elementary decision procedure for model checking pCTL* over generalized discrete Markov processes.

There is a body of past work on applying formal methods to stochastic systems. An early approach is [14] which computes the expected time for a probabilistic program to terminate. Programs are modeled as Markov chains; the techniques are standard in Markov chain theory. [9] describes an approach to the control of finite state Markov processes subject to ω-regular specifications. Criterion for optimality are developed; finding an optimal control strategy reduces to solving a linear program. The approach taken to measuring regular sets is similar in spirit to a sub-procedure in our pCTL* model checking algorithm.

The approaches most related to our research are those based on Temporal Logic. In many past temporal logic formulations of probabilistic verification, a property holds of the system if the measure of acceptable behavior is one; thus in some sense these approaches are *qualitative*. In such paradigms, it is proved that verification can be reduced to performing conventional temporal logic analysis on a model with *fairness constraints* that eliminate events of measure zero; the actual probabilities of events is immaterial. Examples of such approaches are given in [13, 16, 20, 1]. [16] introduces the notion of *α-fairness*; this construct embodies the fact that if a given state is visited infinitely often, then with probability one each state which the given state can make a transition to with non-zero probability is visited infinitely often. [20] follows an approach in which non-deterministic behavior is differentiated from stochastic behavior. Again, verifying such systems reduces to conventional PLTL checking on a Kripke structure with appropriate fairness constraints, which can be done by automata theoretic techniques. [1] verifies *timed* stochastic systems; again in this approach randomness is modelled by fairness. Verification is performed on a fair timed automaton; the proof of correctness is non trivial due to the density of the underlying model of time.

Formulations of quantitative approaches to probabilistic verification also exist. [12] describes an extension of CTL, referred to as PCTL, capable of expressing numerical bounds on the probability of specified properties; it is essentially identical to the logic pCTL defined in § 3. [8] describes an algorithm for determining the measure of a set of paths in a discrete Markov process satisfying a PLTL [10] formula; this immediately provides a procedure for model checking the

logic pCTL*, defined in § 3. Definitions of probabilistic simulation/bisimulation exist in the CCS/CSP community eg [7], which are essentially the same as ours. However, the relationship between probabilistic bisimulation and the expressiveness of probabilistic temporal logic has not been explored in the past.

2 Definitions

2.1 Finite Markov Processes

Definition 1. A *Markov process* (MP) is a 4-tuple $(\mathrm{AP}, S, T, \mathcal{L})$ where AP is a finite set of *atomic propositions*, S is a countable set of *states*, $T : S \times S \rightarrow [0,1]$ is the *transition probability matrix* satisfying the condition $(\forall s \in S)\, \Sigma_{s' \in S} T(s,s') = 1$, and $\mathcal{L} : S \rightarrow 2^{\mathrm{AP}}$ is a *labeling function*. Define $\Sigma = 2^{AP}$ to be the *alphabet* of the FMP. Then each state is labelled by a unique symbol of the alphabet. A *behavior* of the FMP is any subset of Σ^ω. A *strongly connected component* (SCC) is a maximal set of states $\{u_1, u_2, \ldots, u_l\}$ such that for each (u_i, u_j) there is a finite sequence $[v_1 v_2 \ldots v_m]$ such that $T(u_i, v_1) > 0$, $T(v_m, u_j) > 0$, and $T(v_i, v_{i+1}) > 0$, i.e. an SCC is a maximal set of states all which have non-zero probability of reaching each other. The SCC's partition the states; their is a natural directed acyclic graph induced on the SCC's. An SCC having no successor in this graph will be referred to as a *terminal* SCC (tSCC).

Throughout this paper we will restrict our attention to Markov processes in which the set of states is finite (FMPs). Furthermore, the transition probabilities must be rationals expressed as ordered pairs of binary encoded integers. The limitation to rationals simplifies the exposition; it is not a serious restriction in either theory or practise.

Conceptually, an FMP can be viewed as a graph, with weighted directed edges between states, where the weight denotes the probability of the transition, and labels on the state denoting the propositions true at the state. An infinite sequence of states $s_0 s_1 s_2 \ldots$ will be referred to as a *path* through the process; $inf(\sigma)$ denotes the set of states occurring infinitely often in σ. The *word* w corresponding to a path σ is an infinite sequence of subsets of AP where $[w]_k = \mathcal{L}([\sigma]_k)$.

Technically, with any state s_0 in an FMP M we associate a a natural *probability space* $\mathcal{P}_M^{s_0} = (U^{s_0}, \mathcal{C}^{s_0}, \mu^{s_0})$, where the set of all infinite state sequences starting at s_0 is the *universe* U^{s_0}, and the Borel sigma field on U^{s_0} gives the associated space of *events* $\mathcal{C}^{s_0}$, i.e. the class of subsets of U^{s_0} to which probabilities can be assigned. The transition probability matrix T yields the probability measure $\mu^{s_0} : \mathcal{C}^{s_0} \rightarrow [0,1]$; by the measure extension theorem [19], $\mu^{s_0} : \mathcal{C}^{s_0} \rightarrow [0,1]$ is well defined. Given a set β of sequences over the alphabet 2^{AP}, we will abuse notation and refer to the probability of β when we mean the probability of the set of all state sequences starting at s_0 which give words in β. All the sets of state sequences we will be defining later on will be readily seen to be events, i.e. elements of $\mathcal{C}^{s_0}$. We will not dwell on the technicalities of measure theory.

3 pCTL* and pCTL

Syntax and Semantics

There are two type of formulae in pCTL and pCTL*: state formulae (which are true or false in a specific state), and path formulae (which are true or false along a specific path). Let AP be the set of atomic proposition names. A state formula is given by the following syntax:

1. **a** if $a \in AP$
2. If f_1 and f_2 are state formula, then so are $\neg f_1, f_1 \vee f_2$
3. If g is a path formula, then so are $Pr_{<c}(g)$, $Pr_{>c}(g)$, where c is a rational number between 0 and 1 expressed as the ratio of two binary coded integers.

A path formula is given by the following syntax:

1. A state formula
2. If g_1 and g_2 are path formula, then so are $\neg g_1, f_1 \vee g_2$.
3. If g_1 and g_2 are path formula, then so are $Xg_1, g_1 U g_2$.

pCTL* is the set of state formulae that are generated by the above rules; a subset pCTL of pCTL* can be defined in which the path formulae are restricted to be:

1. If f_1, and f_2 are state formula, then then Xf_1 and $f_1 U f_2$ are path formula

Given a finite Markov process $M = (S, AP, \mathcal{L}, T)$ state and path formulae are interpreted as defined below. The formulae f_1 and f_2 are state formulae, and g_1 and g_2 are path formulae. Let s be an arbitrary state, and π be an arbitrary path in M. We now define satisfaction of a state formula with respect to s and path formulae with respect to π.

1. $s, M \models \mathbf{a}$ if and only if $a \in \mathcal{L}(s_0)$
2. $s, M \models \neg f_1$ if and only if $s, M \not\models f_1$, $s, M \models f_1 \vee f_2$ if and only if $s, M \models f_1$ or $s, M \models f_2$
3. $s, M \models Pr_{<c}(g_1)$ if and only if $\mu^s(\{\sigma \in S^\omega \mid \sigma \models g_1 \wedge [\sigma]_0 = s\}) < c$; similarly define $s, M \models Pr_{>c}(g_1)$
4. $\pi, M \models f_1$ if and only if $[\pi]_0, M \models f_1$
5. $\pi, M \models \neg g_1$ if and only if $\pi, M \not\models g_1$, $\pi, M \models g_1 \vee g_2$ if and only if $\pi, M \models g_1$ or $\pi, M \models g_2$
6. $\pi, M \models Xg_1$ if and only if $\pi^1, M \models g_1$, $\pi^1, M \models g_1 U g_2$ if and only if there exists a $k \geq 0$ such that $\pi^k, M \models g_2$ and for all $0 \leq j < k$, $\pi^j, M \models g_1$

When the context is clear, we will write $s \models f_1$ rather than $s, M \models f_1$.

pCTL/pCTL* Model Checking

Lemma 2. Model checking of CTL* over finite Markov processes is decidable in PSPACE.

This result follows immediately from the results of [8] where a PSPACE procedure is given for computing the probability that a given discrete Markov process satisfies a specification in linear logic, specifically PLTL [10]. The procedure is based on obtaining a deterministic ω-automaton accepting the language of the linear specification, and composing it with the Markov process – the measure of the set of accepting states of this composed structure can then be obtained by using accumulation techniques [18]. It should be noted that pCTL can be model checked in polynomial time.

4 Bisimulation for finite Markov processes

Developing a notion of bisimulation is integral to the analysis of state based dynamical systems. Notions of probabilistic bisimulation exists in CCS/CSP literature [7]. In this section we develop a notion of bisimulation for Markov processes and show that it is sound with respect to pCTL*, and furthermore, that pCTL* is expressive with respect to the bisimulation.
Notation: Given an equivalence relation $\mathcal{E} \subset S \times S$, $\mathcal{C}^{\mathcal{E}} = \{C_1, C_2, \ldots, C_l\}$ is the corresponding partition of S. We will write $T(s, C_i)$ for $\Sigma_{t \in C_i}(T(s,t))$.

Definition 3 Bisimulation. Let $M = (AP, S, T, \mathcal{L})$ be a finite Markov process. Define the relation $\mathcal{E} \subset S \times S$ to be a *probabilistic bisimulation* if it is an equivalence relation and

$$\mathcal{E}(s,t) \Rightarrow (\mathcal{L}(s) \equiv \mathcal{L}(t)) \wedge (\forall C_i \in \mathcal{C}^{\mathcal{E}})\,[T(s, C_i) = T(t, C_i)]$$

Define the relation $\mathcal{E}^{prob}$ to be the union of all probabilistic bisimulations. From the fact that there always exists a probabilistic bisimulation, (namely the identity) and closure of probabilistic bisimulations under union, it follows that $\mathcal{E}^{prob}$ is the maximal probabilistic bisimulation. We will refer to states being *probabilistically bisimilar* if they are equivalent under $\mathcal{E}^{prob}$.

Remark: The definition of bisimulation given above is a relation on states in a *single* process; it can trivially be extended to define a relation on processes with designated initial states.

Theorem 4. The probabilistic bisimulation $\mathcal{E}^{prob}$ is sound and complete, i.e.

$$\mathcal{E}^{prob}(s,t) \Longleftrightarrow (\forall \phi \in \text{pCTL}^*)\; s \models \phi \leftrightarrow t \models \phi$$

We break the proof into two lemmas:

Lemma 5 Soundness. States which are probabilistically bisimilar agree in their truth values on all formulae, i.e.

$$\mathcal{E}^{prob}(s,t) \Rightarrow (\forall \phi \in \text{pCTL}^*)\, s \models \phi \leftrightarrow t \models \phi$$

Proof. We use induction on the length of the formula, where the length of $Pr_{<c}(\phi)$ = 1+ length of ϕ. More specifically, our induction hypothesis is the following:
Induction Hypothesis: (IH) On all state formulae ϕ of length $< k$, $\mathcal{E}(s,t) \Rightarrow (s \models \phi \leftrightarrow t \models \phi)$, and on all path formulae ψ of length $< k$,

$$\mathcal{E}(s,t) \Rightarrow (\mu^s(\{\pi \mid ([\pi]_0 = s) \wedge \pi \models \psi\}) = \mu^t(\{\pi \mid ([\pi]_0 = t) \wedge \pi \models \psi)\})$$

Base Case:
The only state and path formulae of length 1 are the atomic propositions. By definition of probabilistic bisimulation, the labels of bisimilar states agree. Hence bisimilar states agree on state formulae of length 1; similarly the IH holds of path formulae of length 1.
Induction:
First we show the result for state formulae. Let the IH hold for all formulae of length less than k. Let ϕ be any state formula of pCTL* of length k.

- $\phi = \neg\phi_1 \mid \phi_1 \wedge \phi_2$, where ϕ_1, ϕ_2 are state formulae: Follows immediately form the IH.
- $\phi = Pr_{<c}(\psi_1) \mid Pr_{>c}(\psi_1)$, where ψ_1 is a path formula: The inductive step follow immediately from the observation that by the induction hypothesis, the measure of the set of paths starting at s satisfying ψ_1 equals the measure of the set of paths satisfying ψ_1 starting at t.

Now we turn our attention to path formulae. Let ψ be any state formula of pCTL*.

- $\psi = \phi$, where ϕ is a state formula: By the above, $s \models \phi \leftrightarrow t \models \phi$. Hence the measure of the set of satisfying paths is 1/0 depending on whether or not the state satisfies ϕ, and induction goes through immediately.
- $\psi = X\psi_1$, where ψ_1 is a path formula: By definition,

  $$\mu^s(\{\pi \mid \pi \models X\psi_1 \wedge [\pi]_0 = s\}) = \sum_{\alpha \in S} (T(s,\alpha) \cdot \mu^\alpha(\{\pi \mid \pi \models \psi_1 \wedge [\pi]_0 = \alpha\}))$$

  By the induction hypothesis all bisimilar states agree on measures of ψ_1; thus the evaluation $\mu^{C_i}(\psi_1)$ (the measure under μ^α of paths starting at a state α in C_i satisfying ψ_1) is well defined. Hence the above can be rewritten to obtain

  $$\mu^s(\{\pi \mid \pi \models X\psi_1 \wedge [\pi]_0 = s\}) = \sum_{C_i \in C^\mathcal{E}} T(s,C_i) \cdot \mu^{C_i}(\psi_1)$$

  It follows by the definition of bisimulation that $(\forall C_i)\,[T(s,C_i) = T(t,C_i)]$; the other terms in the expression are common, and so induction goes through.

- $\psi = \psi_1 U \psi_2$, where ψ_1, ψ_2 are path formulae: Recursively define the path formulae $\theta_0, \theta_1, \ldots$:

$$\theta_0 = \psi_2$$
$$\theta_{n+1} = \psi_1 \wedge X(\theta_n)$$

Define the set of paths A_n^s as

$$A_n^s = \{\pi \mid \pi \models \theta_n \wedge [\pi]_0 = s\}$$

Define $B_0^s = A_n^s$ and $B_{n+1}^s = A_n^s \backslash B_n^s$. Observe that

$$\{\pi \mid \pi \models \psi \wedge [\pi]_0 = s\} = \bigcup_{n \in \omega} B_n^s$$

Since by construction the B_i^s's are disjoint, it follows from elementary analysis that

$$\mu^s(\{\pi \mid \pi \models \psi \wedge [\pi]_0 = s\}) = \sum_{i=0}^{\infty} \mu_s(B_n^s)$$

By induction on n it can be seen that
$(\forall n)\,[\mu^s(B_n^s) = \mu^t(B_n^t)]$; hence $\sum_{i=0}^{\infty}[\mu_s(B_n^s)] = \sum_{i=0}^{\infty}[\mu_s(B_n^t)]$. Thus

$$\mu^s(\{\pi \mid \pi \models \psi \wedge [\pi]_0 = s\}) = \mu^t(\{\pi \mid \pi \models \psi \wedge [\pi]_0 = t\})$$

■

Lemma 6 Completeness. States which agree in their truth values on all formulae are probabilistically bisimilar i.e.

$$(\forall \phi \in \text{pCTL}^*)\, s \models \phi \leftrightarrow t \models \phi \Rightarrow \mathcal{E}^{prob}(s,t)$$

The proof of this lemma proceeds in the spirit of [6]: inductively build up pCTL* formula eventually characterizing states upto $\mathcal{E}^{prob}$ equivalence; details are given in [2].

It should be noted that the states can in fact be characterized in pCTL; this is surprising given that bisimulation under fairness can not be characterized by CTL [3], and that probabilistic transitions are in some loose sense "fair" transitions. However, there is no contradiction – quite simply, arbitrary fairness can not be injected into a system by making certain random probabilistic, and the fairness class which be achieved through randomness is characterizable by pCTL.

5 Generalized Markov Processes

In this section we analyze a larger class of models, namely the generalized Markov processes (GMPs). A GMP consists of a family of Markov processes; the motivation for studying them is that it allows us to develop notions of abstraction and refinement (one GMP containing another) and parametrization (e.g. allowing a transition probability to take a range of values).

Definition 7. A *generalized Markov process* (GMP) G is a triple $(AP, S, \mathcal{L})$ (as given in the definition for Markov processes) and a finite set of constraints on the transition probabilities.

We inductively define the syntax of constraints as follows:

- *Constants* - rational numbers represented as the ratio of binary coded integers
- *Terms* - for each $(s, s') \in S \times S$ introduce a variable $x_{ss'}$; terms are given by the following syntax:

$$c \mid x_{ss'} \mid (t_1 + t_2) \mid (t_1 \cdot t_2)$$

  where t_1, t_2 are terms and c is a constant.
- *Constraints* - are given by the following syntax:

$$t_1 = t_2 \mid t_1 < t_2 \mid \neg(\phi_1) \mid (\phi_1 \wedge \phi_2)$$

  where t_1, t_2 are terms, and ϕ_1, ϕ_2 are constraints.

A given set of constraints defines a subset of $\Re^{|S \times S|}$ in an obvious way. A GMP G on $(S, AP, \mathcal{L})$ defines a natural class $\mathcal{C}^G$ of *compatible* Markov processes in an obvious way; a Markov process $M = (AP, S, T, \mathcal{L})$ compatible with G if it is defined on the same states with identical labelling, and the elements of the transition probability matrix lie in the region defined by the constraints of G (when appropriately indexed).

Definition 8. Let G be a GMP, s a state in G, and ϕ a formula of pCTL*. Then define

$$s, G \models \phi \text{ if and only if } (\forall M \in \mathcal{C}^G)\ s, M \models \phi$$

Note that the resultant logic is non-standard. In particular, the following is not a validity: $s, G \models \phi \vee s, G \models \neg(\phi)$. This is not of any great concern.

Theorem 9. Model checking of CTL* over generalized Markov processes is decidable.

Proof.(Sketch:)

First we make the observation that for a given GMP G, if the associated constraints force $x_{s_i s_j}$'s to zero and the others to be non zero, then there is a unique SCC DAG which is common to all Markov processes in $\mathcal{C}^G$. Secondly,

when the SCC DAG of a Markov process is known, the probability of the set of paths starting at a designated state remaining almost always in a specified SCC is a rational function of $x_{s_i s_j}, s_i, s_j \in S$; this follows from the fact that the probability is the ratio of determinants whose entries are polynomial in the variables $x_{s_i s_j}$.

The two observations can be coupled to obtain a *symbolic* model checking procedure: given a pCTL* formula ϕ and a state s in a GMP G = $(S, AP, \mathcal{L})$ with constraints which uniquely identify the SCC DAG of compatible Markov processes, there is a formula $\kappa_s^\phi(x_{11}, \ldots, x_{nn})$ in the language of Real Closed Fields (RCF) which defines the set of "satisfying assignments" i.e. the set of all $T : S \times S \to [0,1]$ such that whenever T is compatible with the constraints of G then $s, (S, AP, \mathcal{L}, T) \models \phi$. κ_s^ϕ is derived inductively from the subformulae of ϕ. Thus $s, G \models \phi$ if and only if the sentence $(\forall x_{11}, \ldots, x_{nn})\, \kappa_s^\phi$ is a validity of RCF.

It must be noted that given a GMP G, different Markov processes compatible with G may have different SCC DAGs, depending on whether or not certain transitions have zero/non-zero probabilities. Therefore it is necessary to iterate over all possible SCC DAGs; this can be achieved by model checking the set of GMPs indexed by $\{\delta \mid \delta \in \{0,1\}^{|S|\cdot|S|}\}$ where G_δ is G with constraints $(x_{s_i s_j} > 0)$ if and only if $\delta[i,j] \neq 0$. The union of all Markov processes defined by the GMPs of the two classes is precisely $\mathcal{C}^G$; furthermore, since the SCC DAGs of each GMP in the set is unique and defined by the corresponding δ, symbolic model checking on the G_δ's can be performed.

A detailed analysis, exploiting the elementary nature of RCF [5], shows that the procedure is in fact elementary. The existence of efficient procedures for existential fragments of RCF can be invoked to get better bounds. ∎

Definition 10. Given generalized Markov processes G_1 on $(S_1, AP, \mathcal{L}_1)$ and G_2 on $(S_2, AP, \mathcal{L}_2)$, and states s_1, s_2 in the respective processes, define $(s_1, G_1) \prec_{tr} (s_2, G_2)$ if

$$(\forall M_1 \in \mathcal{C}^{G_1})\,(\exists M_2 \in \mathcal{C}^{G_2})\,[\mu_{M_1}^{s_1} = \mu_{M_2}^{s_2}]$$

Theorem 11. There is a decision procedure for checking $\prec_{tr}$.

Proof.(Sketch:)

We use a characterization of Markov processes in terms of *rational* events. Define an output event β to be rational if it is of the form $b_1 \cdot b_2 \cdot b_3 \cdot \ldots b_k \cdot \{0,1\}^\omega$ where $b_i \in \{0,1\}$. We will refer to k as the length of the rational event. The key observation is that if two n state Markov processes with designated initial states agree on their probability measures of all rational events of length less than $2n+1$, then in fact their measures are identical. The proof of this fact is non-trivial; see [15]. This observation will allow us to reduce deciding $\prec_{tr}$ to deciding validity in RCF.

Let G_1 be a GMP such that all Markov processes compatible with G_1 have identical SCC DAGs. Similarly, let G_2 be a GMP with a unique SCC DAG. Let $P_{G_1}^{s_1}(\mathbf{x}, \beta)$ be the function evaluating the probability measure of the set of

paths whose corresponding output sequence is in β as a function of β and $\mathbf{x}$. Clearly, $P^{s_1}_{G_1}(\mathbf{x}, \beta)$ is a rational function of the $x_{s_i s_j}$; it's degree is a function of k. Let $\zeta_1(\mathbf{x})$ be the predicate expressing that $\mathbf{x}$ satisfies the constraints of G_1. Then $(s_1, G_1) \prec_{tr} (s_2, G_2)$ is equivalent to the conjunction over all β for which $k < 2n + 1$ of the following sentences:

$$(\forall \mathbf{x}) [\zeta_2(\mathbf{x}) \rightarrow (\exists \mathbf{y}) (\zeta_1(\mathbf{y}) \wedge (P^{s_1}_{G_1}(\mathbf{x}, \beta) = P^{s_2}_{G_1}(\mathbf{x}, \beta))]$$

i.e. a sentence in RCF. Thus when the SCC DAGs for G_1, G_2 are fixed, $\prec_{tr}$ is decidable.

When the SCC DAGs defined by G_1, G_2 are not unique, an iteration over all possible SCC DAGS is required. Again this can be formulated so as to yield a sentence in RCF; details are left to the reader. A straightforward complexity analysis can be invoked to show that the procedure is elementary. ∎

6 Conclusion and Future Work

To conclude, we have met the primary objective set forth in the introduction, namely to identify a temporal logic capable of quantitative reasoning about the properties of stochastic systems. The logic pCTL* was shown to exhibit an elementary model checking procedure, and was proven to be expressive with respect to probabilistic bisimulation. Furthermore, we were able to extend the universe of models to a very general class of stochastic systems while retaining an elementary model checking procedure.

On the theoretical front, decision procedures for validity of formulae in pCTL* remain to be explored, as well as an explicit sound and complete axiomatization of the logic. It is not clear though that these would be significantly different from the corresponding results for CTL*. Timing and fairness can also be analyzed in conjunction with probabilities. Several issues related to general Markov processes remain open. We would like to study restricted classes of general Markov processes in which the transition probabilities constraints are simply intervals. Further extensions can be made to the generalized model in the sense that probabilities of transitions may be functions of the past; we would like to develop elementary decision procedures for these extended models. The space of measures on $\{0, 1\}^\omega$ is a Banach space; a topological characterization of the subsets of measures definable by generalized Markov processes remains open.

Practically speaking, we feel that pCTL (or minor extensions capable of expressing infinitely often) should be able to express most properties of interest; since model checking pCTL is polynomial time (and in practise good linear algebra packages exist) the main bottleneck will the complexity introduced by composition (variously referred to as hidden Markov models). BDD based symbolic techniques have been successful as a heuristic for coping with state explosion; analogs of the BDD such as the ADD of [4], may similarly prove successful in dealing with composed Markov processes. The advent of efficient packages for finding roots of multi-variate polynomial systems [11] and fragments of RCF [17] is evidence that model checking general Markov processes may be feasible.

References

1. R. Alur, C. Courcoubetis, and D. Dill. Model Checking for Probabilistic Real Time Systems. In *Proc. of the Colloquium on Automata, Languages, and Programming*, pages 115–126, 1991.
2. A. Aziz, V. Singhal, F. Balarin, R. K. Brayton, and A. Sangiovanni-Vincentelli. *It Usually Works: The Temporal Logic of Stochastic Systems.* `http://www-cad.eecs.berkeley.edu:80/ãdnan/cav95.1`.
3. A. Aziz, V. Singhal, F. Balarin, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. Equivalences for Fair Kripke Structures. In *International Colloquium on Automata, Languages and Programming*. Springer Verlag, July 1994.
4. R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Algebraic Decision Diagrams and their Applications . In *Proc. Intl. Conf. on Computer-Aided Design*, pages 188–192, 1993.
5. M. Ben-Or, D. Kozen, and J. Reif. The Complexity of Elementary Algebra and Geometry. In *Proc. ACM Symposium on the Theory of Computing*, pages 457–464, 1984.
6. M. C. Browne, E. M. Clarke, and O. Grümberg. Characterizing Kripke Structures in Temporal Logic. Technical Report CMU-CS-87-104, Department of Computer Science, Carnegie Mellon University, 1987.
7. R. Cleavland, S. A. Smolka, and A. Zwarico. Testing Preorders for Probabilistic Processes. In *Proc. of the Colloquium on Automata, Languages, and Programming*, pages 708–719, 1992.
8. C. Courcoubetis and M. Yannakakis. Verifying Temporal Properties of Finite State Probabilistic Programs. In *Proc. IEEE Symposium on the Foundations of Computer Science*, pages 338–345, 1988.
9. C. Courcoubetis and M. Yannakakis. Automatic Verification of Finite State Programs. In *Proc. of the Colloquium on Automata, Languages, and Programming*, pages 326–347, 1990.
10. E. A. Emerson. Temporal and Modal Logic. In J. van Leeuwen, editor, *Formal Models and Semantics*, volume B of *Handbook of Theoretical Computer Science*, pages 996–1072. Elsevier Science, 1990.
11. Ioannis Emiris. *Sparse Elimination and Applications in Kinematics.* PhD thesis, University of California at Berkeley, Berkeley, CA, December 1994.
12. H. Hansson and B. Jonsson. A Logic for Reasoning about Time and Reliability. *Formal Aspects of Computing*, 6:512–535, 1994.
13. S. Hart and M. Shamir. Probabilistic Temporal Logics for Finite and Bounded Models. In *Proc. ACM Symposium on the Theory of Computing*, pages 1–13, 1984.
14. S. Hart, M. Sharir, and A. Pneuli. Verification of Probabilistic Programs. *SIAM Journal of Computation*, 13:292–314, 1984.
15. A. Paz. *Introduction to Probabilistic Automata* . Academic-Press, 1971.
16. A. Pnueli and L. Zuck. Probabilistic verification. *Information and Computation*, 103(1):1–29, 1993.
17. Ashutosh Rege. *Efficient Decision Procedures for Fragments of R.C.F (*in preparation*).* PhD thesis, University of California at Berkeley, Berkeley, CA, June 1995.
18. D. Revuz. *Markov Chains.* North-Holland, 1975.
19. H. L. Royden. *Real Analysis* . Macmillan Publishing , 1889.
20. M. Y. Vardi and P. L. Wolper. An Automata-Theoretic Approach to Program Verification. In *Proc. IEEE Symposium on Logic in Computer Science*, pages 332–334, 1986.