Model Checking of Recursive Probabilistic Systems

Kousha Etessami
U. of Edinburgh
kousha@inf.ed.ac.uk

Mihalis Yannakakis
Columbia U.
mihalis@cs.columbia.edu

Abstract

Recursive Markov Chains (RMCs) are a natural abstract model of procedural probabilistic programs and related systems involving recursion and probability. They succinctly define a class of denumerable Markov chains that generalize several other stochastic models, and they are equivalent in a precise sense to probabilistic Pushdown Systems. In this paper, we study the problem of model checking an RMC against an ω-regular specification, given in terms of a Büchi automaton or a Linear Temporal Logic (LTL) formula. Namely, given an RMC A and a property we wish to know the probability that an execution of A satisfies the property. We establish a number of strong upper bounds, as well as lower bounds, both for *qualitative* problems (is the probability $= 1$, or $= 0$?), and for *quantitative* problems (is the probability $\geq p$?, or, approximate the probability to within a desired precision). The complexity upper bounds we obtain for automata and LTL properties are similar, although the algorithms are different.

We present algorithms for the qualitative model checking problem that run in polynomial space in the size $|A|$ of the RMC and exponential time in the size of the property (the automaton or the LTL formula). For several classes of RMCs, including single-exit RMCs (a class that encompasses some well-studied stochastic models, e.g., stochastic context-free grammars) the algorithm runs in polynomial time in $|A|$. For the quantitative model checking problem, we present algorithms that run in polynomial space in the RMC and exponential space in the property. For the class of linearly recursive RMCs we can compute the exact probability in time polynomial in the RMC and exponential in the property. For deterministic automata specifications, all our complexities in the specification come down by one exponential.

For lower bounds, we show that the qualitative model checking problem, even for a fixed RMC, is already EXPTIME-complete. On the other hand, even for simple reachability analysis, we showed in [EY05a] that our PSPACE upper bounds in A can not be improved substantially without a breakthrough on a well-known open problem in the complexity of numerical computation.

1 Introduction

Recursive Markov Chains (RMCs) are a natural abstract model of systems that involve probability and recursion, such as procedural probabilistic programs. Informally, an RMC consists of a collection of finite state component Markov chains (MC) that can call each other in a potentially recursive manner. Each component MC has a set of *nodes* (ordinary states), a set of *boxes* (each of which is mapped to a component MC), a well-defined interface consisting of a set of *entry* and *exit* nodes (the nodes where it may start and terminate), and a set of probabilistic transitions connecting the nodes and boxes. A transition to a box specifies the entry node and models the invocation of the component MC associated with the box; when (and if) the component MC terminates at an exit, execution of the calling MC resumes from the corresponding exit of the box.

RMCs are a probabilistic version of Recursive State Machines (RSMs) ([ABE$^+$05]). RSMs and closely related models like Pushdown Systems (PDSs) have been studied extensively in recent research on model checking and program analysis, because of their applications to verification of sequential programs with procedures ([BEM97]). Recursive Markov Chains subsume, in a certain precise sense, several other well-studied models involving probability and recursion: *Stochastic Context-Free Grammars* (SCFGs), have been extensively studied mainly in natural language processing (NLP) (see [MS99]) as well as biological sequence analysis [DEKM99]. A subclass of SCFGs corresponds to a model of web surfing called *backoff* or *back-button process*, studied in [FKK+]. Stochastic context-free grammars can be modeled by a subclass of RMCs, in particular the class of *1-exit* RMCs, in which all components have one exit. *Multi-Type Branching Processes* (MT-BPs), are an important family of stochastic processes, modeling the stochastic evolution of a population of entities of various types (species), with many applications in a great variety of areas such as biology, population dynamics and many others (see, e.g., [Har63, HJV05, KA02]). As shown in [EY05a], the extinction probabilities of branching processes (the central quantities of interest) can be expressed as the termination probabilities of 1-exit RMCs.

RMCs can be viewed also as a recursive version of ordinary finite state Markov chains, in the same way that RSMs are a recursive version of ordinary finite state machines. Markov chains have been used to model non-recursive probabilistic programs and analyze their properties. Probabilistic models of programs and systems are of interest for several reasons. First, a program may use randomization, in which case the transition probabilities reflect the random choices of the algorithm. Second, we may want to model and analyse a program or system under statistical conditions on its behaviour (e.g., based on profiling statistics or on statistical assumptions), and to determine the induced probability of properties of interest.

We introduced RMCs in ([EY05a]), where we developed some of their basic theory and focused on algorithmic reachability analysis: what is the probability of reaching a given state starting from another? In this paper, we study the more general problem of

model checking an RMC against an ω-regular specification: given an RMC A and an ω-regular property, we wish to know the probability that an execution of A satisfies the property. The techniques we develop in this paper for model checking go far beyond what was developed in [EY05a] for reachability analysis.

General RMCs are intimately related to *probabilistic Pushdown Systems (pPDSs)*, an equivalent model introduced in [EKM04], and there are efficient translations between RMCs and pPDSs ([EY05a]). Thus, our results apply with the same complexity to the pPDS model. There has been recent work on model checking of pPDSs ([EKM04, EKM06, BKS05]). As we shall describe below, our results yield substantial improvements, when translated to the setting of pPDSs, on the best upper and lower bounds known for the complexity of ω-regular model checking of pPDSs.

We now outline the main results in this paper. We consider the two most popular formalisms for the specification of ω-regular properties over words, (non-deterministic) Büchi automata (BA for short) and Linear Temporal Logic (LTL). The automata formalism can express all ω-regular properties, while LTL expresses a (important) proper subset. On the other hand, LTL is a common and more succinct formalism. The complexity results turn out to be similar for the two formalisms (even though automata are more general and LTL is more succinct), but require different algorithms.

We are given an RMC A and a property in the form of a (non-deterministic) Büchi automaton (BA) B, whose alphabet corresponds to (labels on) the vertices of A, or a LTL formula φ whose propositions correspond to properties of (labels on) the vertices of A. Let $P_A(L(B))$ (respectively, $P_A(\varphi)$) denote the probability that an execution of A is accepted by B (resp. satisfies the property φ). The *qualitative* model checking problems are: (1) determine whether almost all executions of A satisfy the property (i.e. is $P_A(L(B)) = 1?$, resp. $P_A(\varphi) = 1?$); this corresponds to B or φ being a desirable correctness property, and (2) whether almost no executions of A satisfy the property (i.e. is $P_A(L(B)) = 0?$, resp. $P_A(\varphi) = 0?$), corresponding to B or φ being an undesirable error property. In the *quantitative* model checking problems we wish to compare $P_A(L(B))$ (or $P_A(\varphi)$) to a given rational threshold p, i.e., is $P_A(L(B)) \geq p?$, or alternatively, we may wish to approximate $P_A(L(B))$ to within a given number of bits of precision. Note that in general the probabilities $P_A(L(B))$, $P_A(\varphi)$ may be irrational and may not even be expressible by radicals [EY05a], and hence they cannot be computed exactly.

We show that for both Büchi automata and LTL specifications, the qualitative model checking problems can be solved with an algorithm that runs in PSPACE in the size $|A|$ of the given RMC and EXPTIME in the size of the property specification (i.e., the size $|B|$ of the given automaton B or the size $|\varphi|$ of the given LTL formula φ). More specifically, in a first phase the algorithm analyzes the RMC A by itself (using PSPACE). In a second phase it analyses further A in conjunction with the property, using polynomial time in A and exponential time in the size of the automaton B or the formula φ. If the property is specified by a deterministic automaton B, then the time is polynomial in B.

For several important classes of RMCs we can obtain better complexity. First, if A is a single-exit RMC then the first phase, and hence the whole algorithm, can be done in polynomial time in A. This result applies in particular to (qualitative) model checking of stochastic context-free grammars and backoff processes. Another class of RMCs that we can model-check qualitatively in polynomial time in A is when the total number of entries and exits in A is bounded (we call them *bounded* RMCs). In terms of probabilistic program abstractions, this class of RMCs corresponds to programs with a bounded number of different procedures, each of which has a bounded number of input/output parameter values. The internals of the components of the RMCs (i.e. the procedures) can be arbitrarily large and complex. A third class of RMCs with efficient model checking is the class of *linear RMCs*, i.e. RMCs with linear recursion.

For quantitative model checking, we show that deciding whether $P_A(L(B)) \geq p$ (resp. $P_A(\varphi) \geq p$) for a given rational $p \in [0,1]$ can be decided in PSPACE in $|A|$, and in EXPSPACE in $|B|$ (resp., $|\varphi|$). For a deterministic automaton B, the space is polynomial in both A, B. For linear RMCs we show that the probability $P_A(L(B))$ or $P_A(\varphi)$ is rational and can be computed exactly in polynomial time in the RMC A and exponential time in the specification B or φ. For A a bounded RMC, and when the property is fixed, there is an algorithm that runs in P-time in $|A|$; however, in this case (unlike the others) the exponent of the polynomial depends on the property. Table 1 summarizes our complexity upper bounds.

For lower bounds, we prove that the qualitative model checking problem, even for a fixed, single entry/exit RMC, is already EXPTIME-complete, both for automata and for LTL specifications. On the other hand, even for reachability analysis, we showed in [EY05a] that our PSPACE upper bounds in A, even for the quantitative 1-exit problem, and the general qualitative problem, can not be improved substantially without a breakthrough on the complexity of the *square root sum* problem, a well-known open problem in the complexity of numerical computation (see Section 2.2).

Related Work.

Model checking of ordinary flat (i.e., non-recursive) finite Markov chains has received extensive attention both in theory and practice (eg. [CY95, Kwi03, PZ93, Var85]). It is known that model checking of a Markov chain A with respect to a Büchi automaton B or a LTL formula φ is PSPACE-complete, and furthermore the probability $P_A(L(B))$ or $P_A(\varphi)$ can be computed exactly in time polynomial in A and exponential in B or φ (see [CY95]). Recursive Markov chains were introduced recently in [EY05a], where we developed some of their basic theory and investigated the termination and reachability problems; we summarize the main results in Section 2.2. Recursion introduces a number of new difficulties that are not present in the flat case. For example, in the flat case, the qualitative problems depend only on the structure of the Markov chain (i.e., which transitions are present) and not on the precise values of the transition probabilities; this is not the case for RMCs and numerical issues have to be dealt with even in the qualitative problem. Furthermore, unlike the flat case, the desired probabilities are

Qualitative:

	reachability	det. Büchi	nondet. Büchi or LTL formula
1-exit	P	P	P in RMC, EXPTIME in property
bounded	P	P	P in RMC, EXPTIME in property
linear	P	P	P in RMC, EXPTIME in property
general	PSPACE	PSPACE	PSPACE in RMC, EXPTIME in property

Quantitative:

	reachability	det. Büchi	nondet. Büchi or LTL formula
1-exit	PSPACE	PSPACE	PSPACE in RMC, EXPSPACE in property
bounded	P	P in RMC for fixed Büchi	P in RMC, for fixed property
linear	P	P	P in RMC, EXPTIME in property
general	PSPACE	PSPACE	PSPACE in RMC, EXPSPACE in property

Figure 1: Complexity of Qualitative and Quantitative problems

irrational and cannot be computed exactly.

The equivalent model of probabilistic Pushdown Systems (pPDS) was introduced and studied in [EKM04, BKS05]. They largely focus on model checking against branching-time properties, but they also study deterministic ([EKM04]) and non-deterministic ([BKS05]) Büchi automaton specifications. There are efficient (linear time) translations between RMCs and pPDSs [EY05a], similar to translations between RSMs and PDSs (see [ABE$^+$05]).

This paper combines, and expands on, the content of our two conference publications [EY05c, YE05] on model checking of Recursive Markov Chains. Those two papers treated separately the case of model checking against ω-regular properties and LTL properties. Our upper bounds for model checking, translated to pPDSs, improve substantially on those obtained in [EKM04, BKS05], by at least an exponential factor in the general setting, and by more for specific classes like single-exit, linear, and bounded RMCs. Specifically, [BKS05], by extending results in [EKM04], show that qualitative model checking for a pPDS and a Büchi automaton can be done in PSPACE in the size of the pPDS and 2-EXPSPACE in the size of the Büchi automaton, while quantitative model checking can be decided in EXPTIME in the size of the pPDS and in 3-EXPTIME in the size of the Büchi automaton. They do not obtain stronger complexity results for the class of pBPAs (equivalent to single-exit RMCs). Also, the class of bounded RMCs has no direct analog in pPDSs, as the total number of entries and exits of an RMC gets lost in translation to pPDSs. The above papers do not address directly LTL specifications.

The rest of this paper is organized as follows. In Section 2 we give the necessary definitions and background on RMCs from [EY05a]. We also indicate how the model

checking problems for stochastic context-free grammars (and backoff processes) reduce to (1-exit) RMCs. In Section 3 we show how to construct from an RMC A a flat "summary" Markov chain M'_A which in some sense summarizes the recursion in the trajectories of A; this chain plays a central role analogous to that of the "summary graph" for Recursive State machines [ABE$^+$05]. In Section 4 we address the qualitative model checking problems for Büchi automata specifications, presenting both upper and lower bounds. In Section 5 we show a fundamental "unique fixed point theorem" for RMCs, which allows us to isolate the termination probabilities of an RMC as the unique solution of a set of constraints. In Section 6 we use this to address the quantitative model checking problem for Büchi automata. Section 7 concerns the qualitative model checking of LTL specifications, and Section 8 quantitative model checking of LTL.

2 Definitions and Background

We will first define formally Recursive Markov Chains and give the basic terminology. Then, in Subsection 2.1 we will recall the definitions of Büchi automata and Linear Temporal Logic, and define formally the qualitative and quantitative model checking problems for RMCs. In Subsection 2.2 we will summarize the basic theory of RMCs and results from [EY05a] regarding reachability and termination. In Subsection 2.3 we describe the reduction of stochastic context-free grammars to 1-exit RMCs, with respect to the model checking problems.

A *Recursive Markov Chain (RMC)*, A, is a tuple $A = (A_1, \ldots, A_k)$, where each *component graph* $A_i = (N_i, B_i, Y_i, En_i, Ex_i, \delta_i)$ consists of:

- A set N_i of *nodes*.
- A subset of *entry* nodes $En_i \subseteq N_i$, and a subset of *exit* nodes $Ex_i \subseteq N_i$.
- A set B_i of *boxes*, and a mapping $Y_i : B_i \mapsto \{1, \ldots, k\}$ that assigns to every box (the index of) one of the components, $A_1, \ldots, A_k$. To each box $b \in B_i$, we associate a set of *call ports*, $Call_b = \{(b, en) \mid en \in En_{Y_i(b)}\}$ corresponding to the entries of the corresponding component, and a set of *return ports*, $Return_b = \{(b, ex) \mid ex \in Ex_{Y_i(b)}\}$, corresponding to the exits of the corresponding component.
- A transition relation δ_i, where transitions are of the form $(u, p_{u,v}, v)$ where:
  1. the source u is either a non-exit node $u \in N_i \backslash Ex_i$, or a return port $u = (b, ex)$ of a box $b \in B_i$,
  2. The destination v is either a non-entry node $v \in N_i \setminus En_i$, or a call port $u = (b, en)$ of a box $b \in B_i$,
  3. $p_{u,v} \in \mathbb{R}_{>0}$ is the transition probability from u to v,

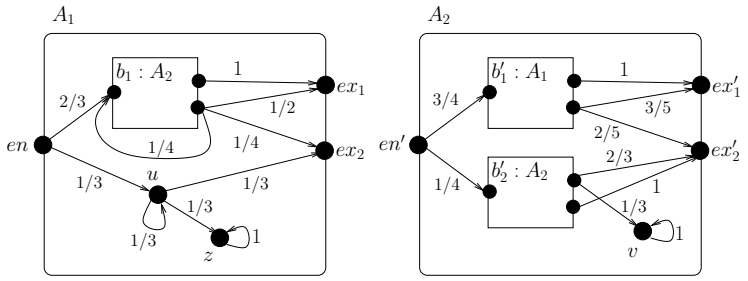

Figure 2: *A sample Recursive Markov Chain*

4. *Consistency of probabilities*: for each u, $\sum_{\{v' | (u, p_{u,v'}, v') \in \delta_i\}} p_{u,v'} = 1$, unless u is a call port or exit node, neither of which have outgoing transitions, in which case by default $\sum_{v'} p_{u,v'} = 0$.

For computational purposes, we assume that the transition probabilities $p_{u,v}$ are rational numbers, given as the ratio of two integers, and we measure their size by the number of bits in the numerator and denominator. The size $|A|$ of a given RMC A is the number of bits needed to specify it (including the size of the transition probabilities).

We will use the term *vertex* of A_i to refer collectively to its set of nodes, call ports, and return ports, and we denote this set by Q_i. Thus, the transition relation δ_i is a set of probability-weighted directed edges on the set Q_i of vertices of A_i. We will use all the notations without a subscript to refer to the union over all the components of the RMC A. Thus, $N = \cup_{i=1}^{k} N_i$ denotes the set of all the nodes of A, $Q = \cup_{i=1}^{k} Q_i$ the set of all vertices, $B = \cup_{i=1}^{k} B_i$ the set of all the boxes, $Y = \cup_{i=1}^{k} Y_i$ the map $Y : B \mapsto \{1, \ldots, k\}$ of all boxes to components, and $\delta = \cup_i \delta_i$ the set of all transitions of A.

An example RMC is shown in Figure 2. The RMC has two components A_1, A_2, each with one entry and two exits (in general different components may have different numbers of entries and exits). Component A_2 has two boxes, b'_1 which maps to A_1 and b'_2 which maps to A_2. Note that the return ports of a box may have different transitions.

An RMC A defines a global denumerable Markov chain $M_A = (V, \Delta)$ as follows. The global *states* $V \subseteq B^* \times Q$ are pairs of the form $\langle \beta, u \rangle$, where $\beta \in B^*$ is a (possibly empty) sequence of boxes and $u \in Q$ is a *vertex* of A. More precisely, the states $V \subseteq B^* \times Q$ and transitions Δ are defined inductively as follows:

1. $\langle \epsilon, u \rangle \in V$, for $u \in Q$. (ϵ denotes the empty string.)
2. if $\langle \beta, u \rangle \in V$ and $(u, p_{u,v}, v) \in \delta$, then $\langle \beta, v \rangle \in V$ and $(\langle \beta, u \rangle, p_{u,v}, \langle \beta, v \rangle) \in \Delta$.
3. if $\langle \beta, (b, en) \rangle \in V$, where $(b, en) \in \mathit{Call}_b$, then $\langle \beta b, en \rangle \in V$ and $(\langle \beta, (b, en) \rangle, 1, \langle \beta b, en \rangle) \in \Delta$.
4. if $\langle \beta b, ex \rangle \in V$, where $(b, ex) \in \mathit{Return}_b$, then $\langle \beta, (b, ex) \rangle \in V$ and $(\langle \beta b, ex \rangle, 1, \langle \beta, (b, ex) \rangle) \in \Delta$.

Item 1 corresponds to the possible initial states, item 2 corresponds to a transition within a component, item 3 corresponds to a recursive call when a new component is entered via a box, item 4 corresponds to the end of a recursive call when the process exits a component and control returns to the calling component.

Some states of M_A are *terminating*, having no outgoing transitions. These are precisely the states $\langle \epsilon, ex \rangle$, where ex is an exit. We want to view M_A as a proper Markov chain, so we consider terminating states to be *absorbing* states, with a self-loop of probability 1.

A *trace* (or *trajectory*) $t \in V^\omega$ of M_A is an infinite sequence of states $t = s_0 s_1 s_2 \ldots$ such that for all $i \geq 0$, there is a transition $(s_i, p_{s_i,s_{i+1}}, s_{i+1}) \in \Delta$, with $p_{s_i,s_{i+1}} > 0$. Let $\Omega \subseteq V^\omega$ denote the set of traces of M_A. For a state $s = \langle \beta, v \rangle \in V$, let $Q(s) = v$ denote the vertex at state s. Generalizing this to traces, for a trace $t \in \Omega$, let $Q(t) = Q(s_0)Q(s_1)Q(s_2)\ldots \in Q^\omega$. We will consider M_A with *initial states* from $Init = \{\langle \epsilon, v \rangle \mid v \in Q\}$. More generally we may have a probability distribution $p_{\text{init}} : V \mapsto [0,1]$ on initial states (we usually assume p_{init} has support only in $Init$, and we always assume it has finite support). This induces a probability distribution on traces generated by random walks on M_A. Formally, we have a probability space $(\Omega, \mathcal{F}, \mathbf{Pr}_\Omega)$, parametrized by p_{init}, where $\mathcal{F} = \sigma(\mathcal{C}) \subseteq 2^\Omega$ is the σ-field generated by the set of *basic cylinder sets*, $\mathcal{C} = \{C(x) \subseteq \Omega \mid x \in V^*\}$, where for $x \in V^*$ the cylinder at x is $C(x) = \{t \in \Omega \mid t = xw,\ w \in V^\omega\}$. The probability distribution $\mathbf{Pr}_\Omega : \mathcal{F} \mapsto [0,1]$ is determined uniquely by the probabilities of cylinder sets, which are given as follows (see, e.g., [Bil95]):

$$\mathbf{Pr}_\Omega(C(s_0 s_1 \ldots s_n)) = p_{\text{init}}(s_0) p_{s_0,s_1} p_{s_1,s_2} \cdots p_{s_{n-1},s_n}$$

We will discuss and obtain improved results for three important classes of RMCs. We say that an RMC is *linearly recursive*, or simply *linear*, if there is no path in any component from a return port of any box to a call port of the same or another box. This corresponds to the usual notion of linear recursion in procedures. For example, the RMC of Fig. 1 is not linear because of the transition from the second exit of box b_1 to the entry of the box; if the transition was not present then the RMC would be linear.

An RMC where every component has at most one exit is called a *1-exit* RMC. As shown in [EY05a], these encompass in a certain sense several well-studied important stochastic models, e.g., Stochastic Context-free Grammars and (Multi-type) Branching Processes, as well as the 'back-button' model of web-surfing studied in [FKK+].

Finally, RMCs where the total number of entries and exits is bounded by a constant c, (i.e., $\sum_{i=1}^{k} |En_i| + |Ex_i| \leq c$) are called *bounded* RMCs. These correspond to recursive programs with a bounded number of different procedures which pass a bounded number of input and output values (the procedures themselves can be internally arbitrarily complicated).

2.1 The central questions for model checking of RMCs.

We first define termination (exit) probabilities that play an important role in our analysis. Given a vertex $u \in Q_i$ and an exit $ex \in Ex_i$, both in the same component A_i, let $q^*_{(u,ex)}$ denote the probability of eventually reaching the state $\langle \epsilon, ex \rangle$, starting at the state $\langle \epsilon, u \rangle$. Formally, we have $p_{\text{init}}(\langle \epsilon, u \rangle) = 1$, and $q^*_{(u,ex)} \doteq \mathbf{Pr}_\Omega(\{t = s_0 s_1 \ldots \in \Omega \mid \exists\, i\,, s_i = \langle \epsilon, ex \rangle\})$. As we shall see, the probabilities $q^*_{(u,ex)}$ will play an important role in obtaining other probabilities.

Two popular formalisms for specifying properties of executions are Büchi automata and Linear Temporal Logic. A *Büchi automaton* (BA for short) $B = (\Sigma, S, q_0, R, F)$, has an alphabet Σ, a set of states S, an initial state $q_0 \in S$, a transition relation $R \subseteq S \times \Sigma \times S$, and a set of accepting states $F \subseteq S$. A *run* of B is a sequence $\pi = q_0 v_0 q_1 v_1 q_2 \ldots$ of alternating states and letters such that for all $i \geq 0$ $(q_i, v_i, q_{i+1}) \in R$. The ω-word associated with run π is $w_\pi = v_0 v_1 v_2 \ldots \in \Sigma^\omega$. The run π is *accepting* if for infinitely many i, $q_i \in F$. Define the ω-language $L(B) = \{w_\pi \mid \pi \text{ is an accepting run of } B\}$. Note that $L(B) \subseteq \Sigma^\omega$. Let $\mathcal{L} : Q \mapsto \Sigma$, be a given Σ-labelling of the vertices of RMC A. $\mathcal{L}$ naturally extends to the state set V of the infinite Markov chain M_A, by letting $\mathcal{L}(\langle \beta, v \rangle) = \mathcal{L}(v)$ for each state $\langle \beta, v \rangle \in V$ of M_A, and it further generalizes to a mapping $\mathcal{L} : V^\omega \mapsto \Sigma^\omega$ from trajectories of M_A, i.e., executions (paths) of the RMC A, to infinite Σ-strings: for $t = s_0 s_1 s_2 \ldots \in V^\omega$, $\mathcal{L}(t) = \mathcal{L}(s_0)\mathcal{L}(s_1)\mathcal{L}(s_2)\ldots$. The execution t *satisfies* the property specified by the automaton B iff $\mathcal{L}(t) \in L(B)$. Given RMC A, with initial state $s_0 = \langle \epsilon, u \rangle$, and given a Büchi automaton B over the alphabet Σ, let $P_A(L(B))$ denote the probability that a trace of M_A is in $L(B)$. More precisely: $P_A(L(B)) \doteq \mathbf{Pr}_\Omega(\{t \in \Omega \mid \mathcal{L}(t) \in L(B)\})$. As in the case of flat (ordinary finite) Markov chains [CY95, Var85], it is easy to show that the sets $\{t \in \Omega \mid \mathcal{L}(t) \in L(B)\}$ are measurable (in $\mathcal{F}$).

Linear Temporal Logic (LTL) [Pnu77] has formulas that are built from a finite set *Prop* of propositions using the usual Boolean connectives (e.g., $\neg, \vee, \wedge$), the unary temporal connective *Next* (denoted $\bigcirc$) and the binary temporal connective *Until* ($\mathcal{U}$); thus, if ξ, ψ are LTL formulas then $\bigcirc\xi$ and $\xi \mathcal{U} \psi$ are also LTL formulas. To specify a property of an RMC using LTL, every vertex of the given RMC A is labelled with a subset of *Prop*: the set of propositions that hold at that vertex. That is, there is a given labelling (often called a *valuation function*) $\mathcal{L} : Q \mapsto \Sigma = 2^{Prop}$. As noted above, the labelling function can be extended naturally to the infinite Markov chain M_A and to its trajectories. If $t = s_0, s_1, s_2 \ldots$ is a trajectory of M_A and φ is an LTL formula, then we define satisfaction of the formula by t at step i, denoted $t, i \models \varphi$ inductively on the structure of φ as follows.

- $t, i \models p$ for $p \in Prop$ iff $p \in \mathcal{L}(s_i)$.
- $t, i \models \neg\xi$ iff not $t, i \models \xi$.
- $t, i \models \xi \vee \psi$ iff $t, i \models \xi$ or $t, i \models \psi$.

- $t, i \models \bigcirc\xi$ iff $t, (i+1) \models \xi$.
- $t, i \models \xi \mathcal{U} \psi$ iff there is a $j \geq i$ such that $t, j \models \psi$, and $t, k \models \xi$ for all k with $i \leq k < j$.

We say that the trajectory t satisfies φ iff $t, 0 \models \varphi$. Other useful temporal connectives can be defined using $\mathcal{U}$. The formula $True \mathcal{U} \psi$ means "eventually ψ holds" and is abbreviated $\Diamond\psi$. The formula $\neg(\Diamond\neg\psi)$ means "always ψ holds" and is abbreviated $\Box\psi$.

If φ is an LTL formula and A is an RMC with a labelling function over the propositions of φ, then the set of executions of A (i.e., trajectories of M_A) that satisfy φ is a measurable set. We use $P_A(\varphi)$ to denote the probability of this set. As is well known, LTL formulas specify ω-regular properties: From a given LTL formula φ over set of propositions $Prop$, one can construct a Büchi automaton B_φ with alphabet $\Sigma = 2^{Prop}$ such that $L(B_\varphi)$ is precisely the set of infinite words that satisfy φ [VW86]. The automaton has in general exponentially larger size than the formula (and this is inherent), i.e., LTL is in general a more succinct formalism. On the other hand, Büchi automata are a more general formalism in that they can express all ω-regular properties, whereas LTL expresses a proper subset.

The *model checking* problems for ω-regular properties of RMCs are defined as follows. We are given a RMC A and a property φ, in terms of either a given LTL formula or a given Büchi automaton B (i.e., $\varphi = L(B)$ in the latter case).

(1) *Qualitative* model checking problems: Is $P_A(\varphi) = 1$? Is $P_A(\varphi) = 0$?

(2) *Quantitative* model checking problems:
a. *Decision problem*: Given a rational $p \in [0, 1]$ (in addition to the RMC A and the property φ), is $P_A(\varphi) \geq p$?
b. *Approximation problem*. Given a number j in unary (in addition to the RMC A and the property φ), approximate $P_A(\varphi)$ to within j bits of precision, i.e., compute a value that is within an additive error 2^{-j} of $P_A(\varphi)$.

Note that if we have a routine for the decision problem $P_A(\varphi) \geq p$?, then we can approximate $P_A(\varphi)$ to within j bits of precision using binary search with j calls to the routine. Thus, for quantitative model checking it suffices to address the decision problem.

Note that probabilistic reachability (and termination) is a special case of model checking for a simple fixed automaton B (or LTL formula φ): Given a vertex u of the RMC A and a subset of vertices F, the probability that the RMC starting at u visits eventually some vertex in F (with some stack context) is equal to $P_A(L(B))$, where we let the labelling $\mathcal{L}$ map vertices in F to 1 and the other vertices of A to 0, and B is the 2-state automaton over alphabet $\{0, 1\}$ that accepts strings that contain a 1. For the termination probability $q^*_{(u,ex)}$, i.e., the probability that the RMC starting at a vertex u terminates at the exit ex of the component A_i of u (with empty stack), let A'

be the RMC obtained from A by adding a new component A'_i that is identical to the component A_i of u; then $q^*_{(u,ex)}$ is equal to the probability that A' starting at vertex u of A'_i reaches the exit ex of A'_i. Similarly, for the *repeated reachability* problem, where we are interested whether a trajectory from u visits infinitely often a vertex of a set F (with any stack context), we can let B be the (2-state deterministic) automaton that accepts strings with an infinite number of 1's. Similarly we can write small fixed LTL formulas for reachability and repeated reachability.

2.2 Basic RMC theory and reachability analysis

We recall some of the basic theory of RMCs developed in [EY05a], where we studied reachability analysis. Considering the termination probabilities $q^*_{(u,ex)}$ as unknowns, we can set up a system of (non-linear) polynomial equations, such that the probabilities $q^*_{(u,ex)}$ are the *Least Fixed Point* (LFP) solution of this system. Use a variable $x_{(u,ex)}$ for each unknown probability $q^*_{(u,ex)}$. We will often find it convenient to index the variables $x_{(u,ex)}$ according to a fixed order, so we can refer to them also as $x_1, \ldots, x_n$, with each $x_{(u,ex)}$ identified with x_j for some j. We thus have a vector of variables: $\mathbf{x} = (x_1\ x_2 \ldots x_n)^T$.

Definition 1 *Given RMC $A = (A_1, \ldots, A_k)$, define the system of polynomial equations, S_A, over the variables $x_{(u,ex)}$, where $u \in Q_i$ and $ex \in Ex_i$, for $1 \leq i \leq k$. The system contains one equation $x_{(u,ex)} = P_{(u,ex)}(\mathbf{x})$, for each variable $x_{(u,ex)}$, where $P_{(u,ex)}(\mathbf{x})$ is a multivariate polynomial with positive rational coefficients. There are 3 cases, based on the "type" of vertex u:*

1. *Type I: $u = ex$. In this case: $x_{(ex,ex)} = 1$.*
2. *Type II: either $u \in N_i \setminus \{ex\}$ or $u = (b, ex')$ is a return port. In these cases:*
$$x_{(u,ex)} = \sum_{\{v|(u,p_{u,v},v)\in\delta\}} p_{u,v} \cdot x_{(v,ex)}.$$
3. *Type III: $u = (b, en)$ is a call port. In this case:*
$$x_{((b,en),ex)} = \sum_{ex' \in Ex_{Y(b)}} x_{(en,ex')} \cdot x_{((b,ex'),ex)}$$

In vector notation, we denote $S_A = (x_j = P_j(\mathbf{x}) \mid j = 1, \ldots, n)$ by: $\mathbf{x} = P(\mathbf{x})$.

Given RMC A, we can construct the system $\mathbf{x} = P(\mathbf{x})$ in polynomial time: $P(\mathbf{x})$ has size $O(|A|\theta^2)$, where θ denotes the maximum number of exits of any component. For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, define $\mathbf{x} \preceq \mathbf{y}$ to mean that $x_j \leq y_j$ for every coordinate j. For $D \subseteq \mathbb{R}^n$, call a mapping $H : \mathbb{R}^n \mapsto \mathbb{R}^n$ *monotone* on D, if: for all $\mathbf{x}, \mathbf{y} \in D$, if $\mathbf{x} \preceq \mathbf{y}$ then $H(\mathbf{x}) \preceq H(\mathbf{y})$. Define $P^1(\mathbf{x}) = P(\mathbf{x})$, and $P^k(\mathbf{x}) = P(P^{k-1}(\mathbf{x}))$, for $k > 1$. Let $\mathbf{q}^* \in \mathbb{R}^n$ denote the n-vector of probabilities $q^*_{(u,ex)}$, using the same indexing as used for $\mathbf{x}$. Let $\mathbf{0}$ denote the all 0 n-vector. Define $\mathbf{x}^0 = \mathbf{0}$, and $\mathbf{x}^k = P(\mathbf{x}^{k-1}) = P^k(\mathbf{0})$, for $k \geq 1$. The map $P : \mathbb{R}^n \mapsto \mathbb{R}^n$ is monotone on $\mathbb{R}^n_{\geq 0}$.

Theorem 1 *([EY05a], see also [EKM04]) The vector of termination probabilities* $\mathbf{q}^* \in [0,1]^n$ *is the Least Fixed Point solution,* $\mathrm{LFP}(P)$*, of* $\mathbf{x} = P(\mathbf{x})$*. Thus,* $\mathbf{q}^* = P(\mathbf{q}^*)$ *and for all* $\mathbf{q}' \in \mathbb{R}^n_{\geq 0}$*, if* $\mathbf{q}' = P(\mathbf{q}')$*, then* $\mathbf{q}^* \preceq \mathbf{q}'$*. Furthermore,* $\mathbf{x}^k \preceq \mathbf{x}^{k+1} \preceq \mathbf{q}^*$ *for all* $k \geq 0$*, and* $\mathbf{q}^* = \lim_{k\to\infty} \mathbf{x}^k$*.*

There are RMCs, even 1-exit RMCs, for which the probability $q^*_{(en,ex)}$ is irrational and not "solvable by radicals" ([EY05a]). Thus, we can't compute probabilities exactly.

Given a system $x = P(x)$, and a vector $q \in [0,1]^n$, consider the following sentence in the *Existential Theory of Reals* (which we denote by $\boldsymbol{ExTh}(\mathbb{R})$):

$$\varphi \equiv \exists x_1, \ldots, x_m \bigwedge_{i=1}^{m} (P_i(x_1, \ldots, x_m) = x_i) \wedge \bigwedge_{i=1}^{m} (0 \leq x_i) \wedge \bigwedge_{i=1}^{m} (x_i \leq q_i)$$

φ is true precisely when there is some $z \in \mathbb{R}^m$, $0 \preceq z \preceq q$, and $z = P(z)$. Thus, if we can decide the truth of this sentence, we could tell whether $q^*_{(u,ex)} \leq p$, for some rational p, by using the vector $q = (1, \ldots, p, 1, \ldots)$. We will rely on decision procedures for $\boldsymbol{ExTh}(\mathbb{R})$. It is known that $\boldsymbol{ExTh}(\mathbb{R})$ can be decided in PSPACE [Can88, Ren92]. Furthermore it can be decided in exponential time, where the exponent depends (linearly) only on the number of variables; thus for a fixed number of variables the algorithm runs in polynomial time. As a consequence:

Theorem 2 *([EY05a]) Given RMC A and rational value* ρ*, there is a PSPACE algorithm to decide whether* $q^*_{(u,ex)} \leq \rho$*, with running time* $O(|A|^{O(1)} \cdot 2^{O(m)})$ *where* m *is the number of variables in the system* $x = P(x)$ *for A. Moreover* $q^*_{(u,ex)}$ *can be approximated to within* j *bits of precision within PSPACE and with running time at most* j *times the above.*

Better results are possible for special classes of RMCs. For linear RMCs, the termination probabilities $q^*_{(u,ex)}$ are rational and can be computed exactly in polynomial time by solving two systems of linear equations. For bounded RMCs, the probabilities are irrational, but it is possible to solve efficiently the quantitative decision and approximation problems by constructing a system of (nonlinear) constraints in a *bounded* number of variables, and using the fact that $\boldsymbol{ExTh}(\mathbb{R})$ is decidable in P-time when the number of variables is bounded. For single-exit RMC the qualitative termination (exit) problem can be solved efficiently. The algorithm does not use the $\boldsymbol{ExTh}(\mathbb{R})$ but rather graph theory and an eigenvalue characterization. We summarize these results in the following theorem.

Theorem 3 *([EY05a])*

1. *For a linear RMC A, the termination probabilities* $q^*_{(u,ex)}$ *are rational and can be computed in polynomial time.*

2. *Given a bounded RMC A and a rational value $p \in [0,1]$, there is a P-time algorithm that decides for a vertex u and exit ex, whether $q^*_{(u,ex)} \geq p$ (or $\leq p$).*

3. *Given a 1-exit RMC A, vertex u and exit ex, we can decide in polynomial time which of the following holds: (1) $q^*_{(u,ex)} = 0$,(2) $q^*_{(u,ex)} = 1$, or (3) $0 < q^*_{(u,ex)}$.*

Hardness, such as NP-hardness, is not known for RMC reachability. However, in [EY05a] we gave strong evidence of "difficulty" in terms of two important open problems: The first one is the *Square-root sum* (SQRT-SUM) problem: given $(d_1, \ldots, d_n) \in \mathbb{N}^n$ and $k \in \mathbb{N}$, decide whether $\sum_{i=1}^n \sqrt{d_i} \leq k$. This problem arises often, especially in geometric computations. It is solvable in PSPACE, but it has been a longstanding open problem since the 1970's whether it is solvable even in NP [GGJ76]. The second problem, called PosSLP ('positive Straight-Line Program'), asks whether a given straight-line program (equivalently, arithmetic circuit) with integer inputs and operations $+, -, *$, computes a positive number or not. It was shown in [ABKPM06] that PosSLP is complete under Cook reductions for the class of decision problems that can be solved in polynomial time in the *unit-cost algebraic RAM* model, a model with unit-cost exact rational arithmetic, i.e., all operations $+, -, *, /$ on rational numbers take unit time, regardless of the size of the numbers. The square-root sum problem can be solved in polynomial time in this model [Tiw92]. Both problems, PosSLP and SQRT-SUM, are in PSPACE (and actually in the Counting Hierarchy [ABKPM06]), but it is not known whether they are in P or even in NP.

In [EY05a] we showed that the PosSLP and SQRT-SUM problems are P-time (many-one) reducible to the quantitative termination problem (i.e. $q^*_{(u,ex)} \geq p$?) for 1-exit RMCs, and to the qualitative termination problem (i.e., $q^*_{(u,ex)} = 1$?) for 2-exit RMCs (see also [BKS05]). Furthermore, even any nontrivial approximation of the termination probabilities (within any additive constant error $c < 1$) for 2-exit RMCs is at least as hard as the PosSLP and SQRT-SUM problems.

As a practical algorithm for numerically computing the probabilities $q^*_{(u,ex)}$, it was proved in [EY05a] that a version of multi-dimensional Newton's method converges monotonically to the LFP of $\mathbf{x} = P(\mathbf{x})$, and constitutes a rapid acceleration of iterating $P^k(\mathbf{0})$, $k \to \infty$.

2.3 Stochastic Context-free Grammars, Backoff Processes, and 1-exit RMCs

A *Stochastic Context-Free Grammar* (SCFG) is a context-free grammar whose rules (productions) have associated probabilities. Formally, a SCFG is a tuple $G = (T, V, R, S_1)$, where T is a set of *terminal* symbols, $V = \{S_1, \ldots, S_k\}$ is a set of *nonterminals*, and R is a set of rules $S_i \xrightarrow{p} \alpha$, where $S_i \in V$, $p \in (0,1]$, and $\alpha \in (V \cup T)^*$, such that for every nonterminal S_i, $\sum_{\langle p_j | (S_i \xrightarrow{p_j} \alpha_j) \in R \rangle} p_j = 1$. S_1 is specified as the starting nonterminal. A

SCFG G generates a language $L(G) \subseteq T^*$ and associates a probability $p(\tau)$ to every terminal string τ in the language, according to the following stochastic process. Start with the starting nonterminal S_1, pick a rule with left hand side S_1 at random (according to the probabilities of the rules) and replace S_1 with the string on the right-hand side of the rule. In general, in each step we have a sentential form, i.e., a string $\sigma \in (V \cup T)^*$; take the leftmost nonterminal S_i in the string σ (if there is any), pick a random rule with left-hand side S_i (according to the probabilities of the rules) and replace this occurrence of S_i in σ by the right-hand side of the rule to obtain a new string σ'. The process stops only when (and if) the current string σ has only terminals. The above process defines a (infinite) Markov chain M_G with state set $(V \cup T)^*$, initial state S_1, and set of terminating states T^*; of course, the unreachable states can be ignored, and also we can add self-loops with probability 1 at the terminating states to make M_G into a proper Markov chain.

The probability $p(\tau)$ of a terminal string $\tau \in T^*$ is the probability that the process reaches (and thus terminates at) the string τ. The above definition of the SCFG process applies a leftmost derivation rule; the probabilities of the terminal strings are the same if one uses any other derivation rule, for example rightmost derivation, or simultaneous expansion in each step of all nonterminals in the current sentential form. The probability of the language $L(G)$ of the SCFG G is $p(L(G)) = \sum_{\tau \in L(G)} p(\tau)$; this is the probability that the stochastic process starting with S_1 generates some terminal string (and terminates).

A probabilistic model of web surfing, called *Random walk with "back buttons"*, or *backoff process*, was introduced and studied in [FKK+]. The model extends an ordinary finite Markov chain with a "back button" feature: There is a finite set of pages (states) $V = \{S_1, \ldots, S_n\}$, and the process starts from some initial page, say S_1. In each step, if the current page is S_i then the process can either proceed along a forward link to a page S_j with probability p_{ij}, or it can 'press the back button' with probability $b_i = 1 - \sum_j p_{ij}$ and return to the previous page from which page S_i was entered. A backoff process C defines an infinite Markov chain M_C on state set V^* with initial state S_1, where each state of M_C is the sequence of pages that led to the current page via forward links. As observed in [EY05a], backoff processes can be mapped to (a subclass of) SCFGs: Given a backoff process C as above, the SCFG G with rules $\{S_i \stackrel{p_{ij}}{\to} S_j S_i | p_{ij} > 0\} \cup \{S_i \stackrel{b_i}{\to} \epsilon | b_i > 0\}$ defines the same infinite Markov chain $M_G = M_C$. Fagin et. al. ([FKK+]) provide a thorough study of backoff processes and efficient algorithms; for example they can approximate in polynomial time to any desired precision the termination probability, i.e. the probability $p(L(G))$ of the language of the associated SCFG. It is an open problem whether such an algorithm exists for the whole class of all SCFGs.

Stochastic context-free grammars (and thus also backoff processes) can be mapped to 1-exit RMCs in a probability-preserving manner [EY05a]: A SCFG G is mapped to a 1-exit RMC A that has one component A_i for each nonterminal S_i of G, the component has one entry en_i and one exit ex_i, and has one path from entry to exit for each rule

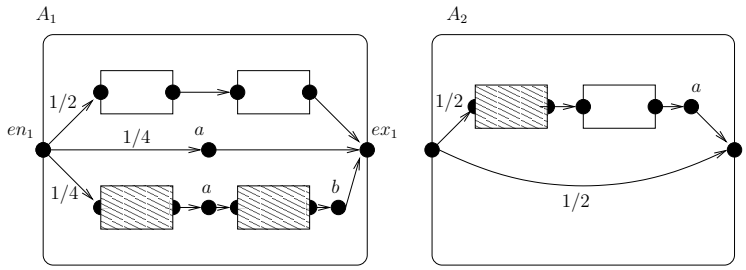

Figure 3: *RMC of a SCFG*

$S_i \xrightarrow{p} \alpha$ of G with left hand side S_i; the path contains a box for every nonterminal on the right-hand side α of the rule mapped to the corresponding component, a node for each terminal in α, the first edge of the path has probability p equal to the probability of the rule and the other edges have probability 1. An example of the mapping is given in Figure 3, which shows the RMC A corresponding to the SCFG G with nonterminals $V = \{S_1, S_2\}$, terminals $T = \{a, b\}$ and rules $R = \{S_1 \xrightarrow{1/2} S_1S_1,\ S_1 \xrightarrow{1/4} a,\ S_1 \xrightarrow{1/4} S_2aS_2b,\ S_2 \xrightarrow{1/2} S_2S_1a,\ S_2 \xrightarrow{1/2} \epsilon\}$. The unshaded boxes of the figure are mapped to A_1 and the shaded boxes are mapped to A_2. All edges that do not have an attached probability label have probability 1.

There is a 1-to-1 correspondence between the trajectories of the infinite Markov chains M_G and M_A associated with a SCFG G and the corresponding RMC A, where the only difference between corresponding trajectories is that the one in M_A executes some additional probability 1 steps.

The mapping from SCFGs to 1-exit RMCs can be used in a straightforward way to reduce model checking questions from SCFGs to 1-exit RMCs. For example, we may consider an execution of the stochastic process of a SCFG G as a sequence of rule applications. Let φ be any ω-regular property over the set R of rules of G, and suppose we wish to know the probability $P_G(\varphi)$ that an execution of G (i.e. a trajectory of M_G) satisfies the property φ. For example, G may be the SCFG for a backoff process C and φ a property concerning the pattern of visits to the pages (states) of C. We can map the SCFG G to a RMC A as above, label the vertices of A that are immediate successors of the entries of the components with the rules corresponding to the edges leading to these vertices, label all the other vertices with some other label i that stands for 'ignore', and let φ' be the property on alphabet $R \cup \{i\}$ obtained from φ by ignoring label i; for example, if φ is given by an automaton B, we add self-loops on letter i to all states of B. It is easy to see then that $P_G(\varphi) = P_A(\varphi')$. Hence, the results we show for 1-exit RMCs apply in particular to SCFGs. Thus for example, the qualitative problems can be solved in polynomial time in the size of the SCFG G and exponential in the size of the property φ (polynomial if φ is given by a deterministic automaton).

A special, finite-string, case of an ω-regular property is the following: given a SCFG $G = (T, V, R, S_1)$ and a regular language (on finite strings) $K \subseteq T^*$, what is the proba-

bility $P_G(K)$ that the SCFG G generates a string in K? The problem can be reduced to a model checking problem for 1-exit RMCs as above, but it is not necessary to use the set R of rules for the labels of the RMC, we can simply use the terminal alphabet T of the SCFG and label the RMC as indicated in the figure. In more detail, assume w.l.o.g that the starting nonterminal S_1 of G does not appear on the right-hand-side of any rule; if it does appear, add a new starting nonterminal S_1' and a rule $S_1' \stackrel{1}{\rightarrow} S_1$. Construct the 1-exit RMC A corresponding to the SCFG G, label the nodes corresponding to terminals in the rules by the terminals as shown in the figure, label the exit of the component of the starting nonterminal by a new 'endmarker' symbol e, label all other vertices by a new symbol i (for 'ignore'), and let K' be the ω-regular language over alphabet $T \cup \{e, i\}$ whose projection to $T \cup \{e\}$ is Ke^ω. Then $P_G(K) = P_A(K')$.

In fact, in the case of finite-string regular language properties K, the model checking problem can be reduced to a termination problem for RMCs. This holds actually more generally, for all RMCs, not only SCFGs. Let A be a labeled RMC (e.g. the 1-exit RMC corresponding to a SCFG G), let B be a deterministic finite automaton (on finite strings) for the language K over the label set of A, and let $P_A(K)$ be the probability, that A generates a terminating trajectory that is in K (e.g., if A is the RMC for a SCFG G, then $P_A(K)$ is the probability $P_G(K)$ that G derives a string in K). From A and B we can construct a (multi-exit) RMC A' of size $|A| \cdot |B|$ (A' is essentially the product of A and B) such that the probability of termination of A' is equal to the probability $P_A(K)$. The RMC A' has generally multiple exits, even if A is a 1-exit RMC (the number of exits is multiplied by $|B|$). For the qualitative problems however, it suffices to deal only with the original RMC A, and thus, if A is a 1-exit RMC, we can solve the qualitative problems in polynomial time in $|A|$ and $|B|$. First, regarding the question '$P_A(K) = 0$?', note that this is equivalent to the question whether A generates any terminating trajectory that is in K; this can be determined in polynomial time by the RSM algorithm of [ABE$^+$05]. (In the special case of a SCFG G, the equivalent question is '$L(G) \cap K = \emptyset$?', which can be tested in polynomial time in the sizes of G and B by standard methods.) Second, regarding the question '$P_A(K) = 1$?', note that this is equivalent to the conjunction of two conditions: (i) the RMC A terminates with probability 1, and (ii) all terminating trajectories are in K, equivalently, there is no terminating trajectory that is accepted by the DFA $\bar{B}$ that accepts the complement of K (in the SCFG case, condition (i) is $p(L(G)) = 1$, and (ii) is $L(G) \cap \bar{K} = \emptyset$.) Condition (ii) can be tested again in polynomial time in $|A|$ and $|B| = |\bar{B}|$ using the RSM algorithms (and by standard methods in the SCFG case). Thus, the question reduces to condition (i), i.e., whether A terminates with probability 1 (whether $p(L(G)) = 1$ in the SCFG case), which can be solved in polynomial time for 1-exit RMCs and SCFGs.

We finish this section with a remark concerning Multi-type Branching Processes [Har63], a classical stochastic model related to SCFGs. We will not give the formal definition here, but we just mention that they involve a finite set of types, corresponding to nonterminals in SCFGs, and they have also a set of probabilistic rules like SCFGs,

except that there are no terminals and the right hand sides of the rules are unordered multi-sets of types (nonterminals) rather than strings. A significant difference in a branching process is that the evolution of the system (i.e., the induced infinite Markov chain) involves in each step a simultaneous expansion of all the types in the current state, rather than a leftmost derivation rule that we used for SCFGs. If we are interested in the probability of termination of the process (called the extinction probability), the derivation rule does not make any difference, and thus the extinction probability can be reduced to the termination probability of a 1-exit RMC [EY05a]. However, if we are interested in other temporal properties of the process, then the derivation rule can matter. Thus, our results in this paper on model checking RMCs do not imply, at least immediately, analogous results for the model checking of more general properties of branching processes.

3 The Conditioned Summary Chain M'_A

For an RMC A, suppose we somehow have the probabilities $q^*_{(u,ex)}$ "in hand". Based on these, we construct a *conditioned summary chain*, M'_A, a finite Markov chain that will will play a key role to model checking RMCs. Since probabilities $q^*_{(u,ex)}$ are potentially irrational, we can not compute M'_A exactly. However, M'_A will be important in our correctness arguments, and we will in fact be able to compute the "structure" of M'_A, i.e., what transitions have non-zero probability. The structure of M'_A will be sufficient for answering various "qualitative" questions.

We will assume, w.l.o.g., that each RMC has one initial state $s_0 = \langle \epsilon, en_{\text{init}} \rangle$, with en_{init} the only entry of some component A_0 that does not contain any exits. Any RMC with any initial node can readily be converted to an "equivalent" one in this form, while preserving relevant probabilities: Given an RMC $A = (A_1, \ldots, A_k)$ with initial node u, which belongs say to component A_i, add a new component A_0 that is a copy of A_i except that it has one new entry node en_{init} which has the same transitions as u, and all the exit nodes of A_i are changed in A_0 into ordinary nodes with probability 1 self-loops.

Before describing M'_A, let us recall from [ABE+05], the construction of a "summary graph", $H_A = (Q, E_{H_A})$, which ignores probabilities and is based only on information about reachability in the underlying RSM of A. Let R be the binary relation between entries and exits of components such that $(en, ex) \in R$ precisely when there exists a path from $\langle \epsilon, en \rangle$ to $\langle \epsilon, ex \rangle$, in the underlying graph of M_A. The edge set E_{H_A} is defined as follows. For $u, v \in Q$, $(u, v) \in E_{H_A}$ iff one of the following holds:

1. u is not a call port, and $(u, p_{u,v}, v) \in \delta$, for $p_{u,v} > 0$.
2. $u = (b, en)$ is a call port, and $(en, ex) \in R$, and $v = (b, ex)$ is a return port.
3. $u = (b, en)$ is a call port, and $v = en$ is the corresponding entry.

For each vertex $v \in Q_i$, let us define the probability of *never exiting*: $\mathrm{ne}(v) = 1 - \sum_{ex \in Ex_i} q^*_{(v,ex)}$. Call a vertex v *deficient* (or a *survivor*) if $\mathrm{ne}(v) > 0$, i.e. there is a nonzero probability that if the RMC starts at v it will never terminate (reach an exit of the component).

We define $M'_A = (Q_{M'_A}, \delta_{M'_A})$ as follows. The set of states $Q_{M'_A}$ of M'_A is the set of deficient vertices: $Q_{M'_A} = \{v \in Q \mid \mathrm{ne}(v) > 0\}$. For $u, v \in Q_{M'_A}$, there is a transition $(u, p'_{u,v}, v)$ in $\delta_{M'_A}$ if and only if one of the following conditions holds:

1. $u, v \in Q$ and $(u, p_{u,v}, v) \in \delta$, and $p'_{u,v} = \frac{p_{u,v} \cdot \mathrm{ne}(v)}{\mathrm{ne}(u)}$. We call these *ordinary transitions*.

2. $u = (b, en) \in \mathit{Call}_b$ and $v = (b, ex) \in \mathit{Return}_b$ and $q^*_{(en,ex)} > 0$, and $p'_{u,v} = \frac{q^*_{(en,ex)} \mathrm{ne}(v)}{\mathrm{ne}(u)}$. We call these *summary transitions*.

3. $u = (b, en) \in \mathit{Call}_b$ and $v = en$, and $p'_{u,v} = \frac{\mathrm{ne}(v)}{\mathrm{ne}(u)}$. We call these transitions, from a call port to corresponding entry, *nesting transitions*.

Note that in all three cases, $p'_{u,v}$ is well-defined (the denominator is nonzero) and it is positive. Recall that we assumed that the initial vertex en_{init} is the entry of a component A_0, and A_0 has no exits. Thus for all $v \in Q_0$, $\mathrm{ne}(v) = 1$, and thus $Q_0 \subseteq Q_{M'_A}$, and if $(u, p_{u,v}, v) \in \delta_0$, then $(u, p_{u,v}, v) \in \delta_{M'_A}$.

Proposition 4 *Probabilities on transitions out of each state in $Q_{M'_A}$ sum to 1.*

Proof. We split into cases.

Case 1: u is any vertex in $Q_{M'_A}$ other than a call port. In this case, $\sum_v p'_{u,v} = \sum_v \frac{p_{u,v} \mathrm{ne}(v)}{\mathrm{ne}(u)}$. Note that $\mathrm{ne}(u) = \sum_v p_{u,v} \mathrm{ne}(v)$. Hence $\sum p'(u, v) = 1$.

Case 2: Suppose u is a call port $u = (b, en)$ in A_i, and box b is mapped to component A_j. Starting at u, the trace will never exit A_i iff either it never exits the box b (which happens with probability $\mathrm{ne}(en)$) or it exits b through some return vertex $v = (b, ex)$ and from there it does not manage to exit A_i (which has probability $q^*_{(en,ex)} \mathrm{ne}((b, ex))$). That is, $\mathrm{ne}((b, en)) = \mathrm{ne}(en) + \sum_{ex \in Ex_j} q^*_{(en,ex)} \mathrm{ne}((b, ex))$. Dividing both sides by $\mathrm{ne}((b, en))$, we have $1 = \mathrm{ne}(en)/\mathrm{ne}((b, en)) + \sum_{ex} q^*_{(en,ex)} \mathrm{ne}((b, ex))/\mathrm{ne}((b, en))$, which is the sum of the probabilities of the edges out of $u = (b, en)$. ∎

M'_A is an ordinary (flat) finite Markov chain. Let $(\Omega', \mathcal{F}', \mathbf{Pr}_{\Omega'})$ denote the probability space on traces of M'_A. We now define a mapping $\rho : \Omega \mapsto \Omega' \cup \{\star\}$, that maps every trace t of the original (infinite) Markov chain M_A, either to a unique trajectory $\rho(t) \in \Omega'$ of the MC M'_A, or to the special symbol $\star$. Trajectories mapped to $\star$ will be

precisely those that go through missing vertices $u \in Q$ that are not in $Q_{M'_A}$, i.e., with $\mathrm{ne}(u) = 0$. We will show that the total probability of all these trajectories is 0, i.e., that $\mathbf{Pr}_\Omega(\rho^{-1}(\star)) = 0$, and moreover, that M'_A preserves the probability measure of M_A: for all $D \in \mathcal{F}'$, $\rho^{-1}(D) \in \mathcal{F}$, and $\mathbf{Pr}_{\Omega'}(D) = \mathbf{Pr}_\Omega(\rho^{-1}(D))$.

We define ρ in two phases. We first define, as a precursor to $\rho(t)$, a map $\rho^H : \Omega \mapsto Q^\omega$, where every trajectory $t \in \Omega$ is mapped to an infinite path $\rho^H(t)$ in the summary graph H_A. Thereafter, we let $\rho(t) = \rho^H(t)$ if all vertices of $\rho^H(t)$ are in M'_A, and let $\rho(t) = \star$ otherwise. We define ρ^H for a trace $t = s_0 s_1 \ldots s_i \ldots$, sequentially based on prefixes of t, as follows. By assumption, $s_0 = \langle \epsilon, en_{\text{init}} \rangle$. ρ^H maps s_0 to en_{init}. Suppose $s_i = \langle \beta, u \rangle$, and, inductively, suppose that ρ^H maps $s_0 \ldots s_i$ to $e_{\text{init}} \ldots u$. First, suppose u is not a call port, and that $s_{i+1} = \langle \beta, v \rangle$; then $s_0 \ldots s_i s_{i+1}$ maps to $e_{\text{init}} \ldots uv$. Next, suppose $u = (b, en)$ is a call port and $s_{i+1} = \langle \beta b, en \rangle$. If the trace eventually returns from this call, i.e. there exists $j > i+1$, such that $s_j = \langle \beta b, ex \rangle$ and $s_{j+1} = \langle \beta, (b, ex) \rangle$, and such that each of of the states $s_{i+1} \ldots s_j$, have βb as a prefix of the call stack, then $s_0 \ldots s_j$ is mapped by ρ^H to $en_{\text{init}} \ldots u(b, ex)$. If the trace never returns from this call, then $s_0 \ldots s_i s_{i+1}$ maps to $en_{\text{init}} \ldots u\ en$. This concludes the definition of ρ^H. We show that the mapping ρ is measure preserving. We start by showing that the trajectories that map to $\star$ have negligible probability:

Lemma 5 $\mathbf{Pr}_\Omega(\rho^{-1}(\star)) = 0$.

Proof. Let $D = \rho^{-1}(\star)$. We can partition D according to the first failure. For $t \in D$, let $\rho^H(t) = w_0 w_1 \ldots \in Q^\omega$. Let $i \geq 0$ be the least index such that $w_i \in Q_{H_A}$ but $w_{i+1} \notin Q_{H_A}$ (such an index must exist). We call $w' = w_0 \ldots w_{i+1}$ a *failure prefix*. Let $C(w') = \{w \in \Omega' \mid w = w'w'' \text{ where } w'' \in Q^\omega\}$ be the cylinder at w', inside $\mathcal{F}'$. Let $D[w'] = \rho^H(C(w'))$.

We claim $\mathbf{Pr}_\Omega(D[w']) = 0$ for all such "failure" prefixes, w'. (To be completely formal, we have to first argue that $D[w'] \in \mathcal{F}$, but this is not difficult to establish: $D[w']$ can be shown to be a countable union of cylinders in $\mathcal{F}$.)

By definition, $\mathrm{ne}(w_i) > 0$, but $\mathrm{ne}(w_{i+1}) = 0$. We distinguish cases, based on what type of vertex w_i and w_{i+1} are.

Case 1: Suppose $w_i \in Q$ is not a call port. In this case, $(w_i, w_{i+1}) \in E_{H_A}$ corresponds to an ordinary edge in the RMC A. A trajectory $t \in D[w']$, is one that reaches $\langle \beta, w_i \rangle$ then moves to $\langle \beta, w_{i+1} \rangle$ and then never exits the component of w_i and w_{i+1}, i.e., retains β as a prefix of the call stack. (This follows from the definition of ρ^H, and the fact that in H_A there are no edges out of exit vertices). Since $\mathrm{ne}(w_{i+1}) = 0$ the probability of such trajectories t is 0, i.e., $\mathbf{Pr}_\Omega(D[w']) = 0$.

Case 2: $w_i = (b, en)$ is a call port, and $w_{i+1} = (b, ex)$. Thus $(w_i, w_{i+1}) \in E_{H_A}$ is a "summary edge", within some component A_k. Again, $\mathrm{ne}(w_i) > 0$, but $\mathrm{ne}(w_{i+1}) = 0$. Any trajectory $t \in D[w']$, reaches $\langle \beta, w_i \rangle$, then sometime later reaches $\langle \beta, w_{i+1} \rangle$, having always retained β as a prefix of the call stack in between, and thereafter it never exits

the component of w_i and w_{i+1}. (Again, similar to case 1, this follows by definition of ρ^H, and H_A.) But since $\mathrm{ne}(w_{i+1}) = 0$, this $\mathbf{Pr}_\Omega(D[w']) = 0$.

Case 3: $w_i = (b, en)$ and $w_{i+1} = en$. In other words, (w_i, w_{i+1}) is an edge of E_{H_A} where we move from a call port to the corresponding entry en of the component A_j, where $Y(b) = j$. Thus a trajectory $t \in D[w']$ enters component A_j at entry en, on step $i+1$, and never exits this component thereafter. Note again, however, that $\mathrm{ne}(w_{i+1}) = 0$. Thus, $\mathbf{Pr}_\Omega(D[w']) = 0$.

Now note that $D = \bigcup_{w'} D[w']$, where the union is over all failure prefixes, $w' \in Q^*$. Note that this is a countable union of sets, each having probability 0, thus $\mathbf{Pr}_\Omega(D) = 0$. ■

Thus, we can effectively ignore trajectories of M_A that are not mapped into trajectories of M'_A. We will now show that the mapping ρ preserves probabilities.

Lemma 6 *For all $D \in \mathcal{F}'$, $\rho^{-1}(D) \in \mathcal{F}$ and $\mathbf{Pr}_\Omega(\rho^{-1}(D)) = \mathbf{Pr}_{\Omega'}(D)$.*

Proof. It suffices, by standard facts about probability measure, to prove the claim for cylinders $C(w') \in \Omega'$, where $w' = w_0, \ldots w_k$. We use induction on k. The base case ($k = 0$) follows from Lemma 5. Namely, $C(\epsilon) = \Omega'$, and $\rho^{-1}(\Omega') = \Omega \setminus \rho^{-1}(\star)$. Thus $\mathbf{Pr}_\Omega(\rho^{-1}(\Omega')) = 1 - \mathbf{Pr}_\Omega(\rho^{-1}(\star)) = 1$.

For the induction step, suppose that the claim holds for the prefix $w' = w_0 w_1 \ldots w_k$. Let $D[w'] = \rho^{-1}(C(w'))$. Define the event $J_{i,y} \in \mathcal{F}$ to be $J_{i,y} = \{t \in \Omega \mid \rho(t) = w_0 \ldots w_i \ldots \text{ , and } w_i = y\}$. Note that by definition of conditional probability, $\mathbf{Pr}_\Omega(D[w'w_{k+1}]) = \mathbf{Pr}_\Omega(D[w'])\,\mathbf{Pr}_\Omega(J_{k+1,w_{k+1}} \mid D[w'])$.

We want to show that $\mathbf{Pr}_\Omega(D[w'w_{k+1}]) = \mathbf{Pr}_{\Omega'}(C(w'w_{k+1}))$. We distinguish three cases, based on what type of edge (w_k, w_{k+1}) is in H_A, as in the proof of Lemma 5.

Case 1: w_k is not a call port. Thus $(w_k, w_{k+1}) \in E_{H_A}$ is an ordinary edge, inside some component A_i of A. Consider the trajectories $t \in D[w'w_{k+1}]$. After some prefix, the trajectory arrives at a vertex $\langle \beta, w_k \rangle$, and subsequently never reaches an exit, i.e., retains β as a prefix of the call stack. The conditional probability $\mathbf{Pr}_\Omega(J_{k+1,w_{k+1}} \mid D[w'])$, is the probability that the $(k+1)$-st step of $\rho(t)$ is w_{k+1}, given that the prefix of $\rho(t)$ is $w_0 w_1, \ldots w_k$. Note that this conditional probability is independent of the call stack β, and that this process has the Markov property, so that it is also independent of how we arrive at w_k. Let $\mathrm{NE}(u) \in \mathcal{F}$ be the event that, starting at a node $\langle \beta, u \rangle$, we will never reach an exit. i.e., $\beta \in B^+$ will forever remain on the call stack.

Since w_k is not a call port, and using the Markovian property, we see that:

$$\begin{aligned}
\mathbf{Pr}_\Omega(J_{k+1,w_{k+1}} \mid D[w']) &= \mathbf{Pr}_\Omega(J_{k+1,w_{k+1}} \mid J_{k,w_k}) \\
&= \mathbf{Pr}_\Omega(J_{1,w_{k+1}} \mid J_{0,w_k}), \text{ (now assuming } p_{\text{init}}(\langle \epsilon, w_k \rangle) = 1) \\
&= \mathbf{Pr}_\Omega(J_{1,w_{k+1}} \cap \text{NE}(w_k)) / \mathbf{Pr}_\Omega(\text{NE}(w_k)) \\
&= \mathbf{Pr}_\Omega(J_{1,w_{k+1}} \cap \text{NE}(w_{k+1})) / \operatorname{ne}(w_k) \\
&= \mathbf{Pr}_\Omega(J_{1,w_{k+1}}) \mathbf{Pr}_\Omega(\text{NE}(w_{k+1})) / \operatorname{ne}(w_k) \\
&= p_{w_k,w_{k+1}} \operatorname{ne}(w_{k+1}) / \operatorname{ne}(w_k)
\end{aligned}$$

Therefore, $\mathbf{Pr}_\Omega(D[w'w_{k+1}]) = \mathbf{Pr}_\Omega(D[w'])p_{w_k,w_{k+1}} \operatorname{ne}(w_{k+1}) / \operatorname{ne} w_k$. By the induction hypothesis, and the construction of M'_A, $\mathbf{Pr}_{\Omega'}(C(w'w_{k+1})) = \mathbf{Pr}_{\Omega'}(C(w'))p'_{w_k,w_{k+1}} = \mathbf{Pr}_\Omega(D[w'])p_{w_k,w_{k+1}} \operatorname{ne}(w_{k+1}) / \operatorname{ne} w_k = \mathbf{Pr}_\Omega(D[w'w_{k+1}])$.

Case 2: $w_k = (b, en)$ is a call port, and $w_{k+1} = (b, ex)$ is a return port. In this case, similar to case 1, we have:

$$\begin{aligned}
\mathbf{Pr}_\Omega(J_{k+1,w_{k+1}} \mid D[w']) &= \mathbf{Pr}_\Omega(J_{1,w_{k+1}} \cap \text{NE}(w_{k+1})) / \operatorname{ne}(w_k), \text{ (assuming } p_{\text{init}}(\langle \epsilon, w_k \rangle) = 1) \\
&= \mathbf{Pr}_\Omega(J_{1,w_{k+1}}) \operatorname{ne}(w_{k+1}) / \operatorname{ne}(w_k) \\
&= q^*_{(en,ex)} \operatorname{ne}(w_{k+1}) / \operatorname{ne}(w_k)
\end{aligned}$$

Again, $\mathbf{Pr}_\Omega(D[w'w_{k+1}]) = \mathbf{Pr}_\Omega(D[w'])q^*_{(w_k,w_{k+1})} \operatorname{ne}(w_{k+1}) / \operatorname{ne}(w_k)$, and by induction, $\mathbf{Pr}_{\Omega'}(C(w'w_{k+1})) = \mathbf{Pr}_{\Omega'}(C(w'))p'_{w_k,w_{k+1}} = \mathbf{Pr}_\Omega(D[w'])q^*_{(w_k,w_{k+1})} \operatorname{ne}(w_{k+1}) / \operatorname{ne}(w_k) = \mathbf{Pr}_\Omega(D[w'w_{k+1}])$.

Case 3: $w_k = (b, en)$ is a call port, and $w_{k+1} = en$ is the corresponding entry. In this case,

$$\begin{aligned}
\mathbf{Pr}_\Omega(J_{k+1,w_{k+1}} \mid D[w']) &= \mathbf{Pr}_\Omega(J_{1,w_{k+1}} \mid J_{0,w_k}) \\
&= \mathbf{Pr}_\Omega(J_{1,w_{k+1}} \cap \text{NE}(w_k)) / \mathbf{Pr}_\Omega(\text{NE}(w_k)), \text{ (assuming } p_{\text{init}}(\langle \epsilon, w_k \rangle) = 1) \\
&= \mathbf{Pr}_\Omega(J_{1,w_{k+1}}) / \operatorname{ne}(w_k), \text{ (because NE}(w_k) \subseteq J_{1,w_{k+1}}) \\
&= \mathbf{Pr}_\Omega(\text{NE}(w_{k+1})) / \operatorname{ne}(w_k) = \operatorname{ne}(w_{k+1}) / \operatorname{ne}(w_k)
\end{aligned}$$

Again, $\mathbf{Pr}_\Omega(D[w'w_{k+1}]) = \mathbf{Pr}_\Omega(D[w']) \operatorname{ne}(w_{k+1}) / \operatorname{ne}(w_k)$, and
$\mathbf{Pr}_{\Omega'}(C(w'w_{k+1})) = \mathbf{Pr}_{\Omega'}(C(w'))p'_{w_k,w_{k+1}} = \mathbf{Pr}_\Omega(D[w']) \operatorname{ne}(w_{k+1}) / \operatorname{ne}(w_k) = \mathbf{Pr}_\Omega(D[w'w_{k+1}])$.
■

Let $H'_A = (Q_{H'_A}, E_{H'_A})$ be the underlying directed graph of M'_A. In other words, the states $Q_{H'_A} = Q_{M'_A}$, and $(u, v) \in E_{H'_A}$ iff $(u, p'_{u,v}, u) \in \delta_{M'_A}$. We will show that we can compute H'_A in P-time for linear RMCs, single-exit RMCs and bounded RMCs, and in PSPACE for arbitrary RMCs. The basic observation is that the structure of M'_A depends only on qualitative facts about the probabilities $q^*_{en,ex}$ and $\operatorname{ne}(u)$, for $u \in Q$.

Proposition 7 *For a RMC A (respectively, linear or single-exit or bounded RMC), and $u \in Q$, we can decide whether* $\mathrm{ne}(u) > 0$ *in PSPACE (respectively, P-time).*

Proof. Suppose u is in a component A_i where $Ex_i = \{ex_1, \ldots, ex_k\}$. Clearly, $\mathrm{ne}(u) > 0$ iff $\sum_{j=1}^{k} q^*_{(u,ex_j)} < 1$. Consider the following sentence, φ, in $\boldsymbol{ExTh}(\mathbb{R})$.

$$\varphi \equiv \exists x_1, \ldots, x_n \bigwedge_{i=1}^{n} (P_i(x_1, \ldots, x_n) = x_i) \wedge \bigwedge_{i=1}^{n} (0 \leq x_i) \wedge \sum_{j=1}^{k} x_{(u,ex_j)} < 1$$

Since $\mathbf{q}^*$ is the LFP solution of $\mathbf{x} = P(\mathbf{x})$, φ is true in the reals if and only if $\sum_{j=1}^{k} q^*_{(u,ex_j)} < 1$. This query can be answered in PSPACE.

For linear RMCs, the termination probabilities can be computed exactly in polynomial time. For single-exit RMCs, we have $Ex_i = \{ex_1\}$, and $\mathrm{ne}(u) > 0$ iff $q^*_{(u,ex_1)} < 1$. As mentioned in section 2.2, this can be answered in P-time for single-exit RMCs ([EY05a]). Similarly, for bounded RMCs the question can be answered in P-time by the techniques developed in [EY05a]. ■

Once we determine the deficient vertices of A, the structure of M'_A can be determined in polynomial time.

Corollary 8 *For a RMC A (respectively, linear, single-exit or bounded RMC), we can compute H'_A in PSPACE (respectively, in polynomial time).*

Proof. Recall that $u \in Q_{H'_A}$ precisely when $u \in Q$ and $\mathrm{ne}(u) > 0$. Thus we can determine the set of nodes with the said complexities, respectively. The transitions of type 1 and 3 in the definition of M'_A are immediately determined. For the type 2 transitions, where $u = (b, en)$ and $v = (b, ex)$, in order to determine whether to include the corresponding summary edge (u, v) we need to decide whether $q^*_{(en,ex)} > 0$. This can be done in polynomial time by invoking the reachability algorithm for RSMs [ABE$^+$05]. ■

4 Qualitative Model Checking for Büchi Automata

We are given a RMC A and a (nondeterministic) Büchi automaton B. To simplify the descriptions of our results, we assume henceforth that the alphabet $\Sigma = Q$, the vertices of A. This is w.l.o.g. since the problem can be easily reduced to this case by relabelling the RMC A and modifying the automaton B (see eg. [CY95]); however care must be taken when measuring complexity separately in the RMC, A, and in the BA, B, since typically B and Σ are small in relation to A. Our complexity results hold with respect to the given inputs A, B.

We will first present our algorithms for qualitative model checking, and then we will prove a lower bound on the complexity of the problem.

4.1 Upper bounds.

Given an RMC $A = (A_1, \ldots, A_k)$ and a (nondeterministic) Büchi automaton $B = (\Sigma, S, q_0, R, F)$ whose alphabet Σ is the vertex set of A, we wish to determine whether $P_A(L(B)) = 1$, $= 0$, or is in-between. We will construct a finite Markov chain $M'_{A,B}$ such that $P_A(L(B))$ is equal to the probability that a trajectory of $M'_{A,B}$ starting from a given initial state reaches one of a designated set of "accepting" bottom SCCs.

First, let $B' = (\Sigma, 2^S, \{q_0\}, R', F')$ be the deterministic automaton obtained by the usual subset construction on B. In other words, the states of B' are subsets $T \subseteq S$, the set F' of accepting states is $F' = \{T | T \subseteq S, T \cap F \neq \emptyset\}$, and the transition function $R' : (2^S \times \Sigma) \mapsto 2^S$ is given by: $R'(T_1, v) = \{q' \in S \mid \exists q \in T_1 \; s.t. \; (q, v, q') \in R\}$. (We are making no claim that $L(B) = L(B')$.)

Next we define the standard *product* RMC, $A \otimes B'$, of the RMC A, and the deterministic Büchi automaton B'. $A \otimes B'$ has the same number of components as A. Call these $A'_1, \ldots, A'_k$. The vertices in component A'_i are pairs (u, T), where $u \in Q_i$ and $T \in 2^S$, and (u, T) is an entry (exit) iff u is an entry (exit). The transitions of A'_i are as follows: there is a transition $((u, T), p_{u,v}, (v, R'(T, v)))$ in A'_i iff there is a transition $(u, p_{u,v}, v)$ in A_i.

Define $M'_{A,B}$ as $M'_{A,B} = M'_{A \otimes B'}$. Thus $M'_{A,B}$ is the conditioned summary chain of RMC $A \otimes B'$. For qualitative analysis on $M'_{A,B}$, we need the underlying graph $H'_{A,B}$. Importantly for the complexity of our algorithms, we do not have to explicitly construct $A \otimes B'$ to obtain $H'_{A,B}$. Observe that states of $M'_{A,B} = (Q \times 2^S, \delta_{M'_{A,B}})$ are pairs (v, T) where v is a state of M'_A, and T a state of B'. The initial state of $M'_{A,B}$ is $(v_0, \{q_0\})$, where v_0 is the initial state of M'_A and q_0 of B. The transitions of $M'_{A,B}$ from a state (v, T) are of three types, corresponding to the types of the transitions out of v in M'_A, as follows:

- Type 1: v is not a call port. Then for every transition $(v, p'_{v,v'}, v') \in \delta_{M'_A}$, we have a corresponding ordinary transition $((v, T), p'_{v,v'}, (v', R'(T, v'))) \in \delta_{M'_{A,B}}$.

- Type 2: v is a call port, $v = (b, en)$. If there is a nesting transition $(v, p_{v,en}, en) \in \delta_{M'_A}$ then there is a nesting transition $((v, T), p_{v,en}, (en, R'(T, en)) \in \delta_{M'_{A,B}}$ with the same probability.

- Type 3: v is a call port, $v = (b, en)$. If v has a summary transition $(v, p_{v,v'}, v')$ in M'_A, where $v' = (b, ex)$, then we have summary transitions of the form $((v, T), p'', (v', T'))$ in $M'_{A,B}$ to states of the form (v', T') iff there exists a path in M_A from $\langle \epsilon, en \rangle$ to $\langle \epsilon, ex \rangle$ which, viewed as a string, drives B' from T to T'; the probability p'' of the transition is $p'' = p' \cdot ne(v')/ne(v)$ where p' is the probability of all such v-v' paths that drive B' from T to T'.

$M'_{A,B}$ is a well-defined Markov chain, which is a refinement of M'_A. That is, every trajectory of $M'_{A,B}$ projected on the first component is a trajectory of M'_A and the

projection preserves probabilities. We can define a mapping σ from the trajectories t of the original (infinite) Markov chain M_A to the trajectories of $M'_{A,B}$, or the special symbol $\star$, in a similar manner as we defined the mapping ρ from trajectories of M to M'_A. For a trajectory t of M_A, it is easy to see that if $\rho(t) \neq \star$ then also $\sigma(t) \neq \star$. Thus, with probability 1 a trajectory of M_A is mapped to one of $M'_{A,B}$. Furthermore, we can show along similar lines the analogue of Lemma 6, i.e. the mapping σ preserves probabilities.

Consider a product graph (without probabilities) $M'_A \otimes B$ between the Markov chain M'_A and the given nondeterministic Büchi automaton B (not B') as follows: The product has nodes (v, q), for all vertices v of M'_A and states q of B, and an edge $(v, q) \to (v', q')$ if either (i) $v \to v'$ is an ordinary edge or a nesting edge of M'_A and q has a transition to q' on input v', or (ii) $v \to v'$ is a summary edge and the RMC has a path from v to v' that corresponds to a run of B from q to q'; if the run goes through an accepting state then we mark the edge $(v, q) \to (v', q')$ as an *accepting* edge. Also, call a node (v, q) *accepting* if $q \in F$ is an accepting state of B.

With every transition (edge) of $M'_{A,B}$ and every edge of $M'_A \otimes B$ we associate a string γ over Σ (the vertex set of A) that caused the edge to be included; i.e., if edge $(v, T) \to (v', T')$ of $M'_{A,B}$ (respectively, edge $(v, q) \to (v', q')$ of $M'_A \otimes B$) corresponds to an ordinary or nesting edge of M'_A then $\gamma = v'$. If it corresponds to a summary edge then we let γ be any string that corresponds to a $v - v'$ path that drives B' from T to T' (resp., for which B has a path from q to q'; if the edge $(v, q) \to (v', q')$ is marked as accepting then we pick a path that goes through an accepting state of B). In the case of a summary edge, there may be many strings γ as above; we just pick anyone of them.

Let t be any trajectory of M_A starting from $\langle \epsilon, v \rangle$, for some vertex v of M'_A and let r be a corresponding run of B starting from a state q. With probability 1, t maps to a trajectory $t' = \rho(t)$ of M'_A. The mapping ρ can be extended to pairs (t, r), where r is a run of B on t, i.e., the pair (t, r) is mapped to a run (path) $r' = \rho(t, r)$ of $M'_A \otimes B$. If r is an accepting run of B then r' goes infinitely often through an accepting node or an accepting edge. The converse does not hold necessarily: a non-accepting run r of B corresponding to a trajectory t may be mapped to a run r' of $M'_A \otimes B$ that traverses infinitely often an accepting edge.

If B is a deterministic Büchi automaton then $M'_{A,B}$ and $M'_A \otimes B$ are clearly the same, except that in $M'_A \otimes B$ we did not include the probabilities of the edges. In this case, the analysis is simpler. Let us say that a bottom strongly connected component (SCC) of $M'_{A,B}$ (and $M'_A \otimes B$) is *accepting* iff it contains an accepting node or an accepting edge.

Theorem 9 *For a RMC A and a deterministic BA B, the probability $P_A(L(B))$ that a trajectory of A is accepted by B is equal to the probability that a trajectory of $M'_{A,B}$ starting from the initial node (v_0, q_0) reaches an accepting bottom SCC.*

Proof. With probability 1 a trajectory t of the RMC A maps to a trajectory $t' = \sigma(t)$

of $M'_{A,B}$ which reaches a bottom SCC C.

If C is not accepting then there is no accepting node or edge in C, hence the run of B on t goes only finitely often through accepting states, and thus t is not accepted by B.

If C is an accepting bottom SCC, then there is an accepting node or an accepting edge in C. If C has an accepting node $(v, q), q \in F$, then with probability 1 the trajectory $t' = \sigma(t)$ of $M'_{A,B}$ goes infinitely often through it, and thus t is accepted by B. Suppose C has an accepting edge $(v, q) \to (v', q')$ and let γ be the string associated with the edge, i.e., γ is a path from v to v' which drives B from q to q' going through an accepting state. With probability 1, a trajectory t whose image $t' = \sigma(t)$ reaches C has the property that t' visits infinitely often (v, q) and furthermore there is an infinite number of such visits where the next substring of t is γ. Thus again, conditioned on the event that t' reaches the bottom SCC C, t is accepted by B with probability 1. ■

Suppose now that B is nondeterministic. We will follow the approach of [CY95] for flat Markov chains, except that here we have to deal with recursive calls and with the summary edges of the constructed Markov chain $M'_{A,B}$ which correspond to sets of paths in the original chain M_A rather than single steps. This complicates things considerably. We will define a set of "special pairs" of the form (v, q), where v is a vertex of M'_A and $q \in F$, which will be useful in characterizing the accepting trajectories.

There are two types of special pairs. The first type is defined as follows. Let v be a vertex of M'_A and $q \in F$ an accepting state of B. Let $D(v, q)$ be the subgraph of $M'_{A,B}$ induced by the node $(v, \{q\})$ and all nodes reachable from it . We say that the pair (v, q) is *special of type 1* if some bottom SCC C of $D(v, q)$ contains a state (v, T) with $q \in T$. We associate with such a pair (v, q) a string $\gamma(v, q) \in \Sigma^*$ that is the concatenation of the strings associated with the edges of $D(v, q)$ on a path from $(v, \{q\})$ to a node of C. (There may be many such paths; just pick anyone.)

The second type of special pair is defined as follows. Let $v = (b, en)$ be a vertex of M'_A that is a call port of a box b of A and let $q \notin F$ be a non-accepting state of B. Define a graph $D(v, q)$ as follows. The graph contains a root node vq and a subgraph of $M'_{A,B}$ consisting of the nodes reachable from vq after we add the following edges. We add an edge from vq to a node $(v', \{q'\})$ of $M'_{A,B}$, where $v' = (b, ex)$ is a return port of the same box b as v, iff there is a path γ from $\langle \epsilon, en \rangle$ to $\langle \epsilon, ex \rangle$ such that B has a run from q to q' on γ that goes through an accepting state; we label the edge $vq \to (v', \{q'\})$ with such a string γ. The graph $D(v, q)$ consists of the root vq and the subgraph of $M'_{A,B}$ induced by all the nodes that are reachable from vq after adding the above edges. We call the pair (v, q) *special of type 2* if some bottom SCC C of $D(v, q)$ contains a state (v, T) with $q \in T$. As in the previous case, we associate with the pair (v, q) a string $\gamma(v, q) \in \Sigma^*$ that is the concatenation of the strings associated with the edges of $D(v, q)$ on a path from vq to a node of C.

Special pairs have the following important properties.

Lemma 10 *Suppose (v,q) is special and that RMC A starts at $\langle \epsilon, v\rangle$ and first performs the transitions in $\gamma(v,q)$. Then with probability 1 such a trajectory t of the RMC is accepted by B with initial state q. Specifically, there is a corresponding accepting run r of B such that $\rho(t,r)$ is a run of $M'_A \otimes B$ starting from (v,q) that infinitely repeats node (v,q) if (v,q) is special of type 1, or repeats an accepting edge out of (v,q) if (v,q) is special of type 2.*

Proof. We construct the accepting run r of B and run r' of $M'_A \otimes B$ one segment at a time. Suppose that (v,q) is special of type 1. Then $\gamma(v,q)$ corresponds to a path in $D(v,q)$ (and $M'_{A,B}$) from $(v,\{q\})$ to a node of a bottom SCC C that contains a state (v,T) with $q \in T$. Consider a trajectory t of the RMC that starts with $\gamma(v,q)$ and the corresponding trajectory t' of $M'_{A,B}$ starting from $(v,\{q\})$. With probability 1, t' exists (i.e. t maps to a trajectory of of $M'_{A,B}$ starting from $(v,\{q\})$), and t' goes to the bottom SCC C and visits infinitely often all the states of C. For every visit to the state (v,T) there is a nonzero probability that in the following steps the trajectory t' will perform the transitions of $\gamma(v,q)$. Hence, with probability 1, at some finite step i, t' visits (v,T) and in the following steps the trajectory t performs $\gamma(v,q)$. Let i be the first time this happens. Since $q \in T$, the prefix of t up to step i has a corresponding run in B from q to q and in $M'_A \otimes B$ from (v,q) to (v,q). This constitutes the first segment of the constructed run r.

At step i, the trajectory t is at vertex v and the suffix from this point on starts again with the sequence $\gamma(v,q)$ of transitions. Since we have a Markov process we can repeat the argument for the remainder of T and construct the second and subsequent segments of r. In general, if E_k denotes the event that the procedure succeeds in constructing k segments, then the probability of E_{k+1} conditioned on E_k is 1. Therefore, the probability of $\cap_k E_k$ is also 1, and thus the required accepting run r will be constructed with probability 1.

Suppose that (v,q) is special of type 2 and let $vq \to (v',\{q'\})$ be the first edge (an accepting edge) in $D(v,q)$ of the path corresponding to $\gamma(v,q)$ that leads from the root vq to the bottom SCC C that contains (v,T) with $q \in T$. Let α be the label of this edge; then $\gamma(v,q) = \alpha\beta$ for some β. The argument is similar to the case of type 1. Consider a trajectory t of the RMC starting from v with the transitions of $\gamma(v,q)$, and let $t = \alpha\tau$. After the prefix α, the trajectory t is at vertex v' (with empty stack, i.e the chain M_A is at vertex $\langle \epsilon, v'\rangle$). The remaining trajectory τ starts with β. With probability 1, τ maps to a trajectory τ' of $M'_{A,B}$ starting from state $(v',\{q'\})$, and since τ starts with β, τ' goes to the bottom SCC C. As in case 1, the trajectory hits with probability 1 infinitely often all the states of C, and furthermore there is a finite time i at which it reaches (v,T) and the following suffix of t starts again with $\gamma(v,q)$. We can map now the prefix of t up to step i to a run of B from q that goes first to q' passing on the way through an accepting state of B (this path corresponds to the prefix α) and then continues and reaches state q again at time i; the corresponding path of $M'_A \otimes B$ follows first the edge to (v',q') and then goes on to reach (v,q). This constitutes the

first segment of the constructed run r. As in case 2, we can then repeat the process to construct the subsequent segments, and the process will succeed with probability 1. ■

Lemma 11 *Suppose there is non-zero probability that a trajectory of the RMC A starting at any vertex $u \in M'_A$ has a corresponding run in $M'_A \otimes B$ starting from any node (u,p) which repeats an accepting state (v,q) infinitely often or repeats an accepting edge $(v,q) \to (v',q')$ infinitely often. Then (v,q) is special.*

Proof. Suppose that an accepting state (v,q) is not special. With probability 1, a trajectory t of the RMC that starts at v corresponds to a trajectory t' of $M'_{A,B}$ that starts at $(v,\{q\})$ and reaches a bottom SCC C of $M'_{A,B}$ (and of $D(v,q)$). Since (v,q) is not special, there is no state (v,T) of C with $q \in T$. Therefore, every run of $M'_A \otimes B$ starting at (v,q) that corresponds to t does not visit (v,q) after t' reaches C, hence, repeats (v,q) only finitely often.

Suppose that t starts at a vertex $u \in M'_A$ and corresponds to a run of $M'_A \otimes B$ starting at a node (u,p) that visits (v,q) infinitely often. Let i be the first step at which the run visits (v,q). The suffix of t from this point on corresponds to a run of $M'_A \otimes B$ starting from (v,q) that visits (v,q) infinitely often. By our above argument, the probability that a trajectory of the RMC has this property is equal to 0, and by the Markov property it follows that the probability that t has such a suffix is also 0.

Consider an accepting edge $(v,q) \to (v',q')$ and suppose that (v,q) is not special. The graph $D(v,q)$ contains an edge $vq \to (v',\{q'\})$. Since (v,q) is not special, no bottom SCC contains any state (v,T) with $q \in T$. Suppose that a trajectory t of the RMC starting at v' corresponds to a run of $M'_A \otimes B$ starting at (v',q') that traverses the edge $(v,q) \to (v',q')$ infinitely often. With probability 1, t corresponds to a trajectory of $M'_{A,B}$ starting from $(v',\{q'\})$ that reaches a bottom SCC C of $D(q,v)$. Since no such bottom SCC contains a state (v,T) with $q \in T$ it follows that every run of $M'_A \otimes B$ from (v',q') that corresponds to t does not visit (v,q) after some point, and hence does not traverse the edge.

Suppose that a trajectory t starts at a vertex $u \in M'_A$ and corresponds to a run of $M'_A \otimes B$ starting at a node (u,p) that visits the edge $(v,q) \to (v',q')$ infinitely often. The argument is similar to the type 1 case. Consider the first time that the edge is traversed and write t as $t = \alpha\tau$, where the prefix α corresponds to the run from (u,p) to (v',q') ending with the traversal of the edge. The suffix τ corresponds to a run starting from (v',q') that repeats the edge infinitely often. From the above argument, the probability that a trajectory τ of the RMC starting at v' has this property is 0, hence the probability that t has such a suffix is also 0. ■

Proposition 12 *$P_A(L(B)) > 0$ iff node (v_0,q_0) in $M'_A \otimes B$ can reach a special node (v,q).*

Proof. Suppose that a trajectory t of the RMC starting at v_0 is accepted by B (starting at q_0), With probability 1, t has a corresponding run in $M'_A \otimes B$ starting at (v_0, q_0) that repeats infinitely often some accepting state (v, q) or some accepting edge $(v, q) \to (v', q')$. It follows from the preceding lemma that (v, q) must be special, and obviously (v_0, q_0) can reach (v, q).

Conversely, suppose that (v_0, q_0) can reach the special pair (v, q) in the graph $M'_A \otimes B$ and let α be the label of such a path from (v_0, q_0) to (v, q). With nonzero probability, the RMC will execute first the sequence of transitions $\alpha\gamma(v, q)$. If this occurs, then from that point on with probability 1 the trajectory will correspond to an accepting run of B. ∎

Call a bottom SCC of the flat Markov chain $M'_{A,B}$ *accepting* if it contains a state (v, T) and T contains some q such that (v, q) is special; otherwise call the bottom SCC *rejecting.*

Theorem 13 *$P_A(L(B))$ is equal to the probability that a trajectory of $M'_{A,B}$ starting from the initial state $(v_0, \{q_0\})$ reaches an accepting bottom SCC.*

Proof. With probability 1 a trajectory t of the RMC maps to a trajectory $t' = \sigma(t)$ of $M'_{A,B}$ which reaches a bottom SCC C.

If C is not accepting then there is no special pair (v, q) such that C contains a state (v, T) with $q \in T$. Then every run of $M'_A \otimes B$ starting from (v_0, q_0) that corresponds to t visits special nodes only finitely many times. It follows that with probability 1, t is not accepted by B.

If C is an accepting bottom SCC, then there is a special pair (v, q) such that C contains a state (v, T) with $q \in T$. The trajectory will visit (v, T) infinitely often, and at every visit there is nonzero probability that the RMC will execute next the sequence $\gamma(v, q)$. Hence, with probability 1 this will occur at some finite point. Then the trajectory t will be accepted by B with probability 1. ∎

It follows that $P_A(L(B)) = 1$ iff all the bottom SCCs of $M'_{A,B}$ reachable from $(v_0, \{q_0\})$ are accepting, and $P_A(L(B)) = 0$ iff no reachable bottom SCC is accepting (or equivalently by Proposition 12, there is no path in $M'_A \otimes B$ from (v_0, q_0) to any special node (v, q)).

As with M'_A and H'_A, let $H'_{A,B}$ denote the underlying directed graph of $M'_{A,B}$. For the qualitative problem, we only need (1) to construct $H'_{A,B}$ and thus only need to know which nodes and edges are present, and (2) to determine which pairs (v, q) are special, and hence which bottom SCCs are accepting. Thus we first have to identify the vertices u of the RMC A for which $\mathrm{ne}(u) > 0$, which can be done in PSPACE for general RMCs, and P-time for single-exit RMCs, linear RMCs, and for bounded RMCs. Then, the edges of $H'_{A,B}$ can be determined by the standard reachability algorithm for RSMs ([ABE$^+$05]). This works by first constructing the genuine product of the underlying

RSM of A (ignoring probabilities on transitions) together with the Büchi automaton B'. This defines a new RSM $A \otimes B'$ (no probabilities), whose size is polynomial in A and B', and thus is exponential in the original non-deterministic Büchi automaton B. The time required for reachability analysis for RSMs is polynomial ([ABE$^+$05]). Thus, once we have identified the deficient vertices of the RMC A, the rest of the construction of $H'_{A,B}$ takes time polynomial in A and B'.

To determine which pairs (v, q) are special, we construct for each candidate pair (v, q) the graph $D(v, q)$. For a pair (v, q) with $q \in F$, this is immediate from $H'_{A,B}$. For a pair (v, q) with $q \notin F$ and $v = (b, en)$ a call port of a box b, we test for each return port $v' = (b, ex)$ of the box and each state q' of B whether there should be an edge $vq \to (v', \{q'\})$; this involves a call to the RSM algorithm of [ABE$^+$05] to determine whether there is a path in the RSM $A \otimes B$ from (en, q) to (ex, q') (with empty stack) that goes through a vertex whose second component is an accepting state of B. Once we determine these edges, we can construct $D(v, q)$. This takes time polynomial in A and B'. Then compute the SCCs of $D(v, q)$, examine the bottom SCCs and check if one of them contains (v, T) with $q \in T$.

Finally, once we have identified the special pairs, we examine the reachable bottom SCCs of $H'_{A,B}$ and determine which ones are accepting and which are rejecting. The dependence of the time complexity on the size of the given RMC A is polynomial except for the identification of the vertices u for which $\mathrm{ne}(u) > 0$. The dependence on $|B|$ is exponential because of the subset construction. If B is deterministic to begin with, we avoid the exponential blow-up and thus have polynomial complexity in B. Thus we have:

Theorem 14 *Given a RMC A and a Büchi automaton B, we can decide whether $P_A(L(B)) = 0$, $P_A(L(B)) = 1$, or $0 < P_A(L(B)) < 1$ in PSPACE in A, and EXPTIME in B. If the given RMC A is a linear, or a bounded, or a 1-exit RMC then the time complexity is polynomial in A. Furthermore, if B is deterministic, the dependence of the time complexity on $|B|$ is also polynomial.*

4.2 Lower Bounds.

We show conversely that the exponential time complexity of qualitative model checking for a nondeterministic Büchi automaton is in general unavoidable.

Theorem 15 *The qualitative problem of determining whether a given RMC A satisfies a property specified by a Büchi automaton B with probability $= 1$, (i.e., whether $P_A(L(B)) = 1$)) is EXPTIME-complete. Furthermore, this holds even if the RMC is fixed and each component has one entry and one exit. Moreover, the qualitative "emptiness" problem, namely determining whether $P_A(L(B)) = 0$, is also EXPTIME-complete, again even when the RMC is fixed and each component has one entry and one exit.*

Proof. We begin by proving the result for determining whether $P_A(L(B)) = 1$ in the case where both A and B are part of the input. The case where A is fixed, and the case for qualitative emptiness, $P_A(L(B)) \stackrel{?}{=} 0$, are variations on the same proof, and we sketch them at the end.

The reduction is from the acceptance problem for alternating linear space bounded Turing machines. As is well known, $\text{ASPACE}(S(n)) = \cup_{c>0}\text{DTIME}(c^{S(n)})$. There is a fixed linear space bounded alternating Turing machine, T, such that the problem of deciding whether T accepts a given input of length n is EXPTIME-complete. We can assume wlog that T has one tape, and uses space n. The tape contains initially the given input x. Recall that an alternating TM has four types of states: existential, universal, accepting and rejecting. We assume wlog that the TM has two possible moves from each existential and universal state, and it halts when it is in an accepting or rejecting state. Let Γ be the tape alphabet, Q the set of states and $\Delta = \Gamma \cup (Q \times \Gamma)$ the extended tape alphabet. A configuration of the TM is expressed as usual as a string of length n where the ith symbol is $(q, X) \in (Q \times \Gamma)$ (we will usually write qX instead of (q, X)) if the head is on the tape cell i, the state is q and the tape symbol is X, and otherwise the ith symbol is the tape symbol X in cell i. The type of a configuration (existential, universal etc) is determined by the type of the state. A *computation* is a sequence of configurations starting from the initial one, according to the transition rules of the TM. We assume wlog that all computations of the TM halt.

There is a natural game associated with an alternating TM between two players, an existential player E and a universal player U. The positions of the game correspond to the configurations. Player E moves at the existential configurations and player U at the universal ones. Accepting configurations are winning positions for player E, and rejecting configurations are winning for player U. An input x is accepted by the TM iff the existential player E has a winning strategy from the initial configuration corresponding to x.

We will construct a RMC, A, and a BA, B, so that A satisfies B with probability 1 iff x is not accepted by T, i.e. E does not have a winning strategy.

Let us first mention that the only thing that will matter about A, is its "structure", i.e., which edges have non-zero probability. We thus describe these edges without defining the probabilities explicitly: any positive probabilities that sum to 1 will do.

The RMC A has an initial component C_0 and a component $C(q, X)$ for each state $q \in Q$ and tape symbol $X \in \Gamma$. The automaton B has an initial state s_0, a final state f which is the only accepting state, and a state (δ, i) for each $\delta \in \Delta$, and $i = 1, \ldots, n$. The alphabet of B is the vertex set of A.

Let q_0 be the initial state of the TM T, and let $x = x_1 \cdots x_n$ be the input. Component C_0 of A has an edge from its entry to a node u_0, an edge from u_0 to a box that is mapped to $C(q_0, x_1)$ and an edge from the exit of the box to an absorbing node v_0 that has a self-loop.

Component $C(q, X)$, where q is an existential state and $X \in \Gamma$, is structured as follows. Suppose that the two moves of the TM T when it is in state q and reads X are $(p_k, Y_k, D_k), k = 1, 2$, where $p_k \in Q$ is the next state, Y_k is the symbol written over X, and $D_k = L/R$ (left/right) is the direction of the head movement. For each $i = 1, .., n$, $k = 1, 2$, and $Z \in \Gamma$, the component has a set of nodes $u[q, X, i, k, Z]$, $v[q, X, i, k, Z]$, and a set of boxes $b[q, X, i, k, Z]$, each mapped to component $C(p_k, Z)$. The entry of the component $C(q, X)$ has edges to each of the nodes $u[q, X, i, k, Z]$, every node $u[q, X, i, k, Z]$ has an edge to the call port of the corresponding box $b[q, X, i, k, Z]$, the return port of each such box has an edge to the corresponding node $v[q, X, i, k, Z]$, and each of these nodes has an edge to the exit of the component.

Component $C(q, X)$, where q is a universal state and $X \in \Gamma$, is structured as follows. Let again the two moves of the TM T for q and X be $(p_k, Y_k, D_k), k = 1, 2$. For each $i = 1, .., n$, $k = 1, 2$, and $Z \in \Gamma$, the component has again a set of nodes $u[q, X, i, k, Z]$, $v[q, X, i, k, Z]$, and a set of boxes $b[q, X, i, k, Z]$ mapped to $C(p_k, Z)$, and has in addition one more node $w[q, X]$. The entry of the component $C(q, X)$ has edges to each of the nodes $u[q, X, i, 1, Z]$, every node $u[q, X, i, 1, Z]$ has an edge to the call port of the corresponding box $b[q, X, i, 1, Z]$, the return port of each such box has an edge to the corresponding node $v[q, X, i, 1, Z]$, and each of these has an edge to node $w[q, X]$. Node $w[q, X]$ has edges to all the nodes $u[q, X, i, 2, Z]$, every node $u[q, X, i, 2, Z]$ has an edge to the call port of the corresponding box $b[q, X, i, 2, Z]$, the return port of each such box has an edge to the corresponding node $v[q, X, i, 2, Z]$, and each of these has an edge to the exit of the component.

Component $C(q, X)$, where q is a halting (accepting or rejecting) state and $X \in \Gamma$ has an edge from its entry to a node $u[q, X]$ and from $u[q, X]$ to the exit of the component.

The transitions of the automaton B are as follows. The initial state s_0 of B transitions on input u_0 to the set of states $\{(q_0 x_1, 1), (x_2, 2), \ldots, (x_n, n)\}$. There are no other transitions out of s_0. The final state f transitions to itself on every input.

Let q be an existential or universal state and suppose that the two moves of the TM T when it is in state q and reads X are $(p_k, Y_k, D_k), k = 1, 2$. On input $u[q, X, i, k, Z]$, a state (δ, j) of B has exactly one transition, as follows: If $j = i$ and $\delta \neq qX$ then it transitions to f; else, if $j = i$ and $\delta = qX$ then it transitions to state (Y_k, i); else, if $((j = i + 1$ and $D_k = R)$ or $(j = i - 1$ and $D_k = L))$ and $\delta = Z$ then it transitions to $(p_k Z, j)$; else, if $((j = i + 1$ and $D_k = R)$ or $(j = i - 1$ and $D_k = L))$ and $\delta \neq Z$ then it transitions to f; else, it transitions to itself, (δ, j). On input $v[q, X, i, k, Z]$, a state (δ, j) of B has the following transition: If $j = i$ then it transitions to (qX, i); else, if $((j = i + 1$ and $D_k = R)$ or $(j = i - 1$ and $D_k = L))$ then it transitions to (Z, j); else, it transitions to itself, (δ, j). All states have a self-loop on input $w[q, X]$, v_0, as well as for all the vertices that are entries and exits of boxes.

Let q be a halting state of the TM. On input $u[q, X]$, a state (δ, j) of B transitions to itself if $\delta \in \Gamma$ or $(\delta = qX$ and q is accepting), and it transitions to f otherwise.

This concludes the definition of the RMC A and the Büchi automaton B. Note that A has a bounded number of components (independent of the length of the input x), and every component has one entry and one exit. Note also that all the transitions of B are deterministic except for the transition of the initial state s_0 on input u_0.

Consider a path of the RMC, and look at the corresponding set P of states of B at each step. At u_0, the set P contains one state (δ, i) for each $i = 1, \ldots, n$ corresponding to the initial configuration of the TM. From then on, it is easy to check that P always contains *at most* one state (δ, i) for each i, and either these states form a configuration of the TM or P contains f. Once f is included in P, then it will stay there forever and any continuation of the path will be accepted by B.

Let us call a path of the RMC *valid* if the set P at the end (and during the path) does not contain f. Consider the game tree G of the game corresponding to the TM T on the given input x: The nodes of the tree are the configurations reached by the TM in its computation, the root is the initial configuration, the children of each node are the two successor configurations, and the leaves correspond to halting configurations. An existential strategy corresponds to a subtree G_E of G that contains one child of each (reachable) existential configuration (nodes that are not reachable any more from the root are not included in G_E). We consider the two children of each node as being ordered according to the indexing ($k = 1, 2$) of the two moves of the configuration.

We claim that every valid path of the RMC corresponds to a prefix of the depth-first-search traversal of an existential game tree G_E, where all the leaves in the prefix are accepting; and conversely every such prefix of a DFS traversal corresponds to a valid path. Note that when a valid path is at the entry of an existential component $C(q, X)$, in order for it to continue to be valid it must move to a node $u[q, X, i, k, Z]$ such that i is the current position of the head, q and X must be the current state and symbol at cell i, and Z must be the symbol in the tape cell where the head moves next according to move $k = 1$ or 2 of the TM. That is, there are precisely two valid choices corresponding to the two possible moves of the existential player. The transitions of B are defined so that the states of the new current set P form the next configuration as the path of the RMC moves to the box corresponding to the move of the TM. When the path exits the box, if it is still valid, then the set P is the same as when the path entered the box. After the node $v[q, X, i, k, Z]$, the set P is updated to restore the configuration as it was when the component $C(q, x)$ was called. For a universal component $C(q, X)$ there is only one correct choice if the path is to remain valid. If the path exits the component remaining valid, it means that it never went through a rejecting component, i.e., the corresponding subtree of G_E that was traversed has only accepting leaves.

If x is accepted by the TM T, then the existential player has a winning strategy, hence there is a valid path of the RMC that reaches node v_0 of C_0 and stays there forever. Thus, with positive probability the RMC follows this path which is not accepted by B. On the other hand, if x is not accepted by the TM T, then every path becomes eventually invalid (either because it reaches a rejecting component or because one of its transitions

does not correspond to a TM move) and hence is accepted by B; thus the probability of acceptance is 1.

We are done with the proof that checking $P_A(L(B)) = 1$ is EXPTIME-hard. By Theorem 14, the problem is also EXPTIME-complete.

We now sketch how a variation of the same proof shows that probabilistic emptiness ($P_A(L(B)) > 0$?) is also EXPTIME-complete.

For each component except C_0, add a direct path from entry to exit $en \to r \to ex$ through a new node r where the first edge has probability $> 1/2$. Every state of the Büchi automaton B, goes to f on these intermediate nodes. (The purpose of these paths is to make sure that every component exits with probability 1 - but these are not valid paths). Remove the self loop of v_0, add new nodes y_0,z_0 to C_0, and edges $v_0 \to y_0 \to z_0 \to u_0$ with probability 1. Also add a new state g to B which is the only accepting state (f is not accepting anymore). On input y_0, all states of B die (have no transition) except for f that goes to g. On z_0, g goes to the initial state s_0.

By the previous proof , (1) if input x is accepted by the TM T, the old RMC had a path p from the initial vertex to v_0 such that the corresponding set of states of the automaton at the end (for all possible runs) did not include f. (2) If x is not accepted by the TM T, then for every trajectory of the old RMC, the automaton has a run that gets to f.

Because of the new paths to the exits that we have added, every component exits with probability 1 (this follows from basic facts about RMCs, see [EY05a]). Hence, infinitely often (i.o.), the trajectory will go to u_0, traverse a path, come out at v_0, go to y_0,z_0, back to u_0, and again all over. If the state set of the Büchi automaton includes f when the path arrives at v_0, then it will go next to g, then reset to the initial state and start again. Therefore, if x is not accepted by the TM T, this will happen every time, hence g will appear i.o. and the probability of acceptance $P_A(L(B) = 1$.

If x is accepted by the TM T, and in some iteration the RMC follows the path p as above then the automaton will die when the path reaches y_0. Every time the process returns to u_0 and tries again, there is positive probability that it will follow the path p, so eventually this will happen at some point with probability 1. When it happens, the automaton will die and hence will not accept the trajectory. Thus, in this case $P_A(L(B)) = 0$.

Next, we briefly sketch how we actually only need a fixed RMC, whose size does not depend on the size of the input tape of the TM. Here is the modification. Drop the tape cell index i from the u and v nodes of A, and add a self loop to these nodes; that is, the u and v nodes have now the form $u[q, X, k, Z], v[q, X, k, Z]$ for $q \in Q, X, Z \in \Gamma, k = 1, 2$. Basically, the RMC is going to guess what is the correct index i of the cell with the tape head, which will be the number of times it loops at the node u (and v). The Büchi automaton states keep track of how many times the RMC goes around the loop at the current vertex $u[q, X, k, Z]$ or $v[q, X, k, Z]$. In other words, the BA states have now, besides extended tape symbol $\delta \in \Delta$ and cell number $i = 1, \ldots, n$, another counter

$j = 0, 1, \ldots, n$ for the number of iterations of the self-loop at the current u or v vertex of the RMC. If the RMC performs the wrong number of iterations at the current vertex (stays too long or leaves too early) then the BA transitions to f and the game is in effect over. In particular if the BA is at state (qX, i, j) and the counter j tries to exceed i without the RMC leaving the current vertex $u[...]$, or if it leaves $u[...]$ before j reaches i, then the the BA goes to f. If the RMC leaves the current vertex $u[...]$ exactly at the correct time, then (qX, i, i) makes the right transition to the appropriate state $(Y, i, 0)$ corresponding to the Turing machine move. For the other states (δ, i, j) of the BA, first if δ has a state and is not qX then go to f right away; otherwise, if the state is (δ, l, i) when the RMC moves out of $u[...]$ and $l \neq i$, the state assumes that the RMC moved at the right time (i.e. tape head is at cell i) and acts accordingly: for example if the head is supposed to move left and new state $= p$, new symbol (in new position)$= Z$, then (δ, l, i) transitions to $(\delta, l, 0)$ if $l \neq i - 1$, to f if $l = i - 1$ but $\delta \neq Z$, and to $(pZ, l, 0)$ otherwise. The moves at $v[...]$ that restore the state are similar.

■

5 The Unique Fixed Point Theorem

As we have mentioned, the transition probabilities of the chain $M'_{A,B}$ are in general irrational and cannot be computed exactly, but instead have to be determined implicitly. To do quantitative model checking in PSPACE in $|A|$, it will be crucial to use $\boldsymbol{ExTh}(\mathbb{R})$ to uniquely identify these probabilities. For this, we need first to have a set of constraints that uniquely identify the termination probabilities of a RMC. These probabilities are the least fixed point of the system $x = P(x)$. However, the system has in general multiple fixed points. We will show in this section that adding a certain set of additional constraints ensures a unique fixed point, the desired LFP(P).

Consider a RMC A. First, we can determine in polynomial time the vertex-exit pairs (u, ex) for each component such that the probability $q^*_{(u,ex)} = 0$. Introduce variables $x_{u,ex}$ only for the remaining pairs. (Alternatively, we could include also variables $x_{(u,ex)}$ for the pairs with 0 probability, and include the equation $x_{(u,ex)} = 0$.) Note that if a vertex u cannot exit its component, i.e. $q^*_{(u,ex)} = 0$ for all ex then there is no variable involving u. Consider the set of fixed point equations $x = P(x)$, where we omit the terms that involved "missing" variables. The least fixed point q^* is the true vector of probabilities of each vertex u reaching exit ex (with empty stack). Recall that a vertex u is called *deficient* (or a survivor) if $\sum_{ex} q^*_{(u,ex)} < 1$, i.e. $ne(u) > 0$; otherwise u is *full*. Note that by the qualitative analysis, we can determine which vertices are deficient and which are full in PSPACE. We will show the following:

Theorem 16 *(The Unique Fixed Point Theorem) The set of equations $x = P(x)$ has a unique nonnegative fixed point that satisfies $\sum_{ex} x_{(u,ex)} < 1$ for every deficient vertex*

u, and $\sum_{ex} x_{(u,ex)} \leq 1$ for every other vertex u. (This fixed point, of course, is $q^* = \mathrm{LFP}(P)$.)

Proof. Suppose that there is another nonnegative fixed point y, besides the least fixed point, that satisfies the constraints on $\sum_{ex} x_{(u,ex)}$. Since q^* is the least fixed point we have $q^* \leq y$. If u is a full vertex then $\sum_{ex} y_{(u,ex)} \leq 1 = \sum_{ex} q^*_{(u,ex)}$ and $q^* \leq y$ imply that $y_{(u,ex)} = q^*_{(u,ex)}$ for every ex.

We will show below that y agrees with q^* also on the deficient vertices. Let (u, ex) be a pair such that $y_{(u,ex)} > q^*_{(u,ex)}$. We will derive a contradiction.

Let $x_{(u,ex)} = f_1(x)$ be the equation for variable $x_{(u,ex)}$ in the system $x = P(x)$. The right hand side $f_1(x)$ is a sum of monomials and possibly a constant term. If u is not a call port then each monomial is of the form $p_{u,v}x_{(v,ex)}$, where v is a successor of u. If $u = (b, en)$ is a call port of a box b then each monomial is of the form $x_{en,ex'}x_{(b,ex'),ex}$ where ex' is an exit of the component corresponding to box b; in the latter case we consider the variables of the monomial as ordered. We will rewrite iteratively the right hand side $f_1(x)$ as follows. In the ith iteration we have an expression $f_i(x)$ which is the sum of a constant term (possibly 0) and of a set of *ordered* monomials; i.e. each monomial has a constant coefficient and the product of a sequence of variables (with possible repetitions allowed) in a specific order. We take every non-constant monomial and replace the leftmost variable of the monomial by the right hand side of its equation in the system $x = P(x)$. We combine like terms (treated again as ordered monomials) and let $f_{i+1}(x)$ be the resulting expression.

Observe first that both fixed points, q^* and y satisfy the equation $x_{(u,ex)} = f_n(x)$ for all n. Second, we claim that $f_n(x)$ is related to the (infinite) Markov chain M_A corresponding to the RMC A in the following way. Let Z_n be the state at time n of the chain M_A with initial state $\langle \epsilon, u \rangle$. Note that if the chain hits $\langle \epsilon, ex \rangle$ at some time t then it stays there forever, i.e. $Z_n = \langle \epsilon, ex \rangle$ for all $n \geq t$.

Lemma 17 *The constant term of $f_n(x)$ is equal to $Prob(Z_n = \langle \epsilon, ex \rangle)$. Furthermore, for each state $\langle \beta, v \rangle$ where $\beta = b_1 \dots b_j$ is a sequence of boxes and v is a vertex such that $Prob(Z_n = \langle \beta, v \rangle) > 0$, and for every sequence $\gamma = w_1, \dots, w_j$ of exits of the components corresponding to the boxes such that the variables with indices $(v, w_j), ((b_j, w_j), w_{j-1}), \dots ((b_2, w_2), w_1), ((b_1, w_1), ex)$ exist, the expression $f_n(x)$ has an ordered monomial $Prob(Z_n = \langle \beta, v \rangle)x_{(v,w_j)}x_{((b_j,w_j),w_{j-1})} \cdots x_{((b_2,w_2),w_1)}x_{((b_1,w_1),ex)}$. If β is the empty string ϵ then the monomial is simply $Prob(Z_n = \langle \epsilon, v \rangle)x_{(v,ex)}$. These are all the monomials of $f_n(x)$.*

Proof. By induction, starting with $f_0(x) = x_{(u,ex)}$. The basis is trivial: $Prob(Z_0 = \langle \epsilon, u \rangle) = 1$. For the induction step, consider a monomial of $f_n(x)$ corresponding to the state $\langle \beta, v \rangle$ and a sequence γ of exits to the boxes (if β is nonempty). If v is an exit and $\beta = \epsilon$, then v must be ex (because for other exits the variable does not exist since it is 0), and $x_{v,ex}$ will be replaced by 1, increasing the constant term. If v is an exit and

$\beta \neq \epsilon$, then v must be w_j (again because otherwise the variable does not exist). In this case we will replace also $x_{(v,w_j)}$ by 1, which corresponds to the chain M_A moving from state $\langle b_1 \dots b_j, v\rangle$ to state $\langle b_1 \dots b_{j-1}, (b_j, w_j)\rangle$, i.e. returning from the call of box b_j to the return port (b_j, w_j).

If v is not a call port (or an exit) then the equation for the leftmost variable $x_{(v,w_j)}$ is $\sum_{v'} p_{v,v'} x_{(v',w_j)}$ where the sum ranges over all successors v' of v for which the variable $x_{(v',w_j)}$ exists. In particular, if $\beta = \epsilon$, then $x_{(v,ex)} = \sum_{v'} p_{v,v'} x_{(v,ex)}$. Note also that $Prob(Z_{n+1} = \langle \beta, v'\rangle | Z_n = \langle \beta, v\rangle) = p_{v,v'}$.

Finally, if $v = (b', v')$ is a call port of a box b' corresponding to some component A_k with an entry v', then we will replace the leftmost variable $x_{(v,w_j)}$ with $\sum_{w'} x_{(v',w')} x_{((b',w'),w_j)}$ where the sum ranges over all exits w' of A_k for which both variables $x_{(v',w')}, x_{((b',w'),w_j)}$ exist. This corresponds to the chain moving with probability 1 from state $\langle \beta, v\rangle$ to state $\langle \beta b', v'\rangle$, and including all feasible extensions $w'\gamma$ of γ. ■

Let N be any fixed positive integer and consider n going to infinity. We can write $f_n(x)$ as the sum of three terms $c_n, g_n(x), h_n(x)$, where $c_n = Prob(Z_n = \langle \epsilon, ex\rangle)$ is the constant term. A monomial $Prob(Z_n = \langle \beta, v\rangle) x_{(v,w_j)} x_{((b_j,w_j),w_{j-1})} \cdots x_{((b_2,w_2),w_1)} x_{((b_1,w_1),ex)}$ corresponding to a state $\langle \beta, v\rangle$, and a sequence $\gamma = w_1, \dots, w_j$ of exits is included in the second term $g_n(x)$ iff at most N of the vertices $v, (b_j, w_j) \dots (b_2, w_2)(b_1, w_1)$ are deficient; otherwise it is included in $h_n(x)$. Clearly, as $n \to \infty$, the first term $c_n \to q^*_{(u,ex)}$. For q^*, the second and third term $g_n(q^*), h_n(q^*)$ tend to 0 as $n \to \infty$, because by definition $q^*_{(u,ex)} = c_n + g_n(q^*) + h_n(q^*)$. Now, consider the two terms $g_n(y)$ and $h_n(y)$.

Let r be the minimum component in q^* (recall, r is positive, because we have removed variables $x_{(u,ex)}$ where $q^*_{(u,ex)} = 0$). Then clearly $y \leq \mathbf{1} \leq q^*/r$ (coordinate-wise inequality). Since in every monomial of the second term, $g_n(x)$, at most N of the vertices are deficient, and since q^* and y have the same value for each index pair whose first component is a full vertex, it follows that the value of each monomial of $g_n(x)$ evaluated at y is bounded from above by the value of the monomial evaluated at q^* divided by r^N. Hence $g_n(y) \leq g_n(q^*)/r^N$. Since N is fixed and $g_n(q^*) \to 0$ as $n \to \infty$, it follows that also $g_n(y) \to 0$ as $n \to \infty$.

Consider all the monomials in the third term $h_n(y)$ corresponding to a state $\langle \beta, v\rangle$ of M_A, and let $\beta = b_1 \cdots b_j$. Let G be the following (ordinary) layered Markov chain: G has a source node v, then it has j layers (numbered from j down to 1) and finally it has a sink node ex. Each layer i contains a node labelled w_i for each exit w_i of the component corresponding to the box b_i. In addition there is a dead state d. Nodes ex and d have self-loops with probability 1. There is a transition from v to a node w_j in layer j with probability $y_{(v,w_j)}$ iff the corresponding variable $x_{(v,w_j)}$ exists. For each pair of nodes w_i, w_{i-1} in successive layers, $i, i-1$ there is a transition from node w_i of layer i to node w_{i-1} of layer $i-1$ with probability $y_{((b_i,w_i),w_{i-1})}$ if the corresponding variable exists. Finally there is a transition from each node w_1 of layer 1 to the sink ex with probability $y_{((b_1,w_1),ex)}$ (if the variable exists). Note that the probabilities of the above

transitions out of a node of G sum to less than 1 iff the corresponding vertex v or (b_i, w_i) of the RMC is deficient. Let D be the set of these 'deficient' nodes of G. For every deficient node add a transition to the dead state d with the missing probability. Let U be the set of deficient vertices of the RMC, and let $p = \min\{1 - \sum_{ex'} y_{(u',ex')} | u' \in U\}$. Note that $p > 0$. Each deficient node of G has a transition to d with probability at least p. We need the following fact about (ordinary) finite Markov chains.

Lemma 18 *Let G be a finite Markov chain, and let D be a subset of states such that each state $u \in D$ has a transition with probability at least $p > 0$ to a dead (absorbing) state d. Then for every positive integer N, the probability that, a trajectory of G starting at any state visits at least N times a state of D and is not absorbed in the dead state d, is at most $(1-p)^N$.*

Proof. Every time the chain visits a state in D it has probability at least p of transitioning to d, and probability at most $1 - d$ of surviving (continuing without being absorbed in d). Hence if it visits D N times then the probability that it is still surviving is at most $(1-p)^N$. We can give a formal proof of this by induction on N. The basis, $N = 0$, is trivial. For the induction step, suppose the claim holds for $N - 1$. Let $E_i(s)$ be the event that G starting from state s survives i visits to D. Then $P(E_N(s)) = \sum_{u \in D} P(u \text{ is the first visited state of } D) P(E_N(u))$. Now, $P(E_N(u)) = \sum_{v \neq d} p_{u,v} P(E_{N-1}(v))$. By induction hypothesis $P(E_{N-1}(v)) \leq (1-p)^{N-1}$ for all v, and $\sum_{v \neq d} p_{u,v} \leq 1 - p$ since $u \in D$. Therefore, $P(E_N(u)) \leq (1-p)^N$, and hence $P(E_N(s)) \leq (1-p)^N$. ∎

By our construction of G, every monomial of $h_n(y)$ involving the state $\langle \beta, v \rangle$ corresponds to a path in G from v to ex that goes through at least N deficient nodes; the value of the monomial is equal to $Prob(Z_n = \langle \beta, v \rangle)$ times the probability of the path in G. The lemma implies then that the contribution to $h_n(y)$ of the set of monomials for state $\langle \beta, v \rangle$ is at most $Prob(Z_n = \langle \beta, v \rangle)(1-p)^N$. Therefore, $h_n(y) \leq (1-p)^N$. Since $(1-p) < 1$ and N is an arbitrary integer, the right hand side can be made arbitrarily small.

Recall the earlier established facts that $c_n \to q^*_{(u,ex)}$ and $g_n(y) \to 0$, as $n \to \infty$. Note also that we must have, for all n, $y_{(u,ex)} = f_n(y) = c_n + g_n(y) + h_n(y)$. Thus note that, for any $\epsilon > 0$, we can pick N and n large enough, with $N \leq n$, such that $f_n(y) \leq q^*_{(u,ex)} + \epsilon$. But if $0 < \epsilon < y_{(u,ex)} - q^*_{(u,ex)}$, then $f_n(y) < y_{(u,ex)}$, which contradicts the fact that $y_{(u,ex)} = f_n(y)$ for all n. Hence $q^*_{(u,ex)} = \lim_{n \to \infty} f_n(y) = y_{(u,ex)}$. ∎

6 Quantitative Model Checking for Büchi Automata

We now provide algorithms for the quantitative model checking of an RMC with respect to a given Büchi automaton.

Theorem 19 *Given a Recursive Markov Chain, A, and Büchi automaton, B, and a rational value $p \in [0,1]$, we can decide whether $P_A(L(B)) \geq p$ in PSPACE in $|A|$ and in EXPSPACE in $|B|$, specifically in space $O(|A|^{c_1} 2^{c_2|B|})$ for some constants c_1, c_2. Furthermore, if B is deterministic we can decide this in PSPACE in both A and B.*

Proof. We make crucial use of Theorem 16, and we combine this with use of the summary chain $M'_{A,B}$, and queries to $\boldsymbol{ExTh}(\mathbb{R})$. Observe that by Theorem 13, all we need to do is "compute" the probability that a trajectory of $M'_{A,B}$, starting from the initial state $(v_0, \{q_0\})$ reaches an accepting bottom SCC. We can not compute $M'_{A,B}$ exactly, however, we will be able to identify the transition probabilities uniquely inside a $\boldsymbol{ExTh}(\mathbb{R})$ query, and will, inside the same query identify the probability of reaching an accepting bottom SCC.

Let $\mathbf{q}^* = \mathrm{LFP}(P)$ be the solution vector of probabilities for the system $\mathbf{x} = P(\mathbf{x})$ associated with RMC A. Recall that by Proposition 7, we can compute in PSPACE in $|A|$ the set $Q' = \{u \in Q \mid \mathrm{ne}(u) > 0\}$ of deficient vertices. We do this as a first step. Consider next the following quantifier-free formula, where $c(u)$ is the index of the component of a vertex u:

$$\varphi_1(\mathbf{x}) \equiv (\mathbf{x} = P(\mathbf{x})) \wedge (0 \preceq \mathbf{x}) \wedge \bigwedge_{u \in Q'} \Big(\sum_{ex \in Ex_{c(u)}} x_{(u,ex)} < 1 \Big) \wedge \bigwedge_{u \in Q \setminus Q'} \sum_{ex \in Ex_{c(u)}} (x_{(u,ex)} = 1)$$

By Theorem 16, the only solution vector $\mathbf{x}$ in $\mathbb{R}^n$ for which $\varphi_1(\mathbf{x})$ holds true is $\mathbf{q}^*$. In other words, φ_1 uniquely identifies $\mathrm{LFP}(P)$.

Recall that $\mathrm{ne}(u) = 1 - \sum_{ex \in Ex_{c(u)}} q^*_{(u,ex)}$. Now, let $\mathbf{y}$ be a vector of variables indexed by vertices of A, and let $\varphi_2(\mathbf{x}, \mathbf{y}) \equiv \bigwedge_{u \in Q} (y_u = 1 - \sum_{ex \in Ex_{c(u)}} x_{(u,ex)})$. The only vector of reals $(\mathbf{x}, \mathbf{y})$ that satisfies $\varphi_1 \wedge \varphi_2$ is the one where $x_{(u,ex)} = q^*_{(u,ex)}$ and $y_u = \mathrm{ne}(u)$.

Recall the construction of $M'_{A,B}$. The states of $M'_{A,B}$ are pairs (v, T), where $v \in Q'$, and $T \subseteq S$ is a set of states of B. The transitions of $M'_{A,B}$ come in three varieties.
Case 1: v is not a call port, and $(v, p'_{v,v'}, v') \in \delta_{M'_A}$. Then we have a corresponding transition $((v,T), p'_{v,v'}, (v', R'(T,v'))) \in \delta_{M'_{A,B}}$, where $p'_{v,v'} = p_{v,v'} \,\mathrm{ne}(v')/\,\mathrm{ne}(v)$, and thus $p'_{v,v'} \,\mathrm{ne}(v) = p_{v,v'} \,\mathrm{ne}(v')$. Associate a variable $z_{v,v'}$ with each such probability $p'_{v,v'}$, and define the formula: $\varphi_3(\mathbf{y}, \mathbf{z}) \equiv \bigwedge_{(v,v') \in \mathrm{Case1}} (z_{v,v'} y_v = p_{v,v'} y_{v'})$.
Case 2: v is a call port, $v = (b, en)$ where v is vertex in component A_i and box b is mapped to component A_j, and $v' = en$, and there is a *nesting* transition $(v, p'_{v,v'}, v') \in \delta_{M'_A}$. Then there is a *nesting* transition $((v,T), p'_{v,v'}, (v', R'(T,v')) \in \delta_{M'_{A,B}}$ with the same probability. Here $p'_{v,v'} = \mathrm{ne}(v')/\,\mathrm{ne}(v)$, and thus $p'_{v,v'} \,\mathrm{ne}(v) = \mathrm{ne}(v')$. Associate a variable $z_{v,v'}$ with each such probability $p'_{v,v'}$, and define: $\varphi_4(\mathbf{y}, \mathbf{z}) \equiv \bigwedge_{(v,v') \in \mathrm{Case2}} (z_{v,v'} y_v = y_{v'})$.
Case 3: v is a call port that has a summary transition $(v, p'_{v,v'}, v')$ in M'_A to a vertex $v' = (b, ex)$. Then we have summary transitions of the form $((v,T), p'', (v', T'))$ in $M'_{A,B}$ to the following set of states of the form (v', T'): If there exists a path of M_A that starts

at the entry en of A_j and ends at the exit ex (with empty call stack) which, viewed as a string drives B' from T to T', then we include the edge $((v,T), p'_{(v,T),(v',T')}, (v',T'))$ in $\delta_{M'_{A,B}}$, where $p'_{(v,T),(v',T')} = q^*_{((en,T),(ex,T'))} \cdot ne(v')/ne(v)$, and where $q^*_{((en,T),(ex,T'))}$ is the probability of reaching $\langle \epsilon, (ex,T') \rangle$ from $\langle \epsilon, (en,T) \rangle$ in the product RMC $A \otimes B'$. First, compute $A \otimes B'$ and its associated equations $\mathbf{w} = P^{\otimes}(\mathbf{w})$ explicitly. Note that $|A \otimes B'| = O(|A||B'|)$. Let $Q^{\otimes}$ be the set of vertices of $A \otimes B'$. We can compute the set $Q'^{\otimes}$ of vertices v of $A \otimes B'$, for which $\mathrm{ne}(v) > 0$ in PSPACE in $|A \otimes B'|$. Consider now the quantifier-free formula:

$$\varphi_5(\mathbf{w}) \equiv (\mathbf{w} = P^{\otimes}(\mathbf{w})) \wedge (0 \preceq \mathbf{w}) \wedge \bigwedge_{u \in Q'^{\otimes}} \Big(\sum_{ex \in Ex_{\mathrm{c}(u)}} w_{(u,ex)} < 1 \Big) \wedge \bigwedge_{u \in Q^{\otimes} \setminus Q'^{\otimes}} \Big(\sum_{ex \in Ex_{\mathrm{c}(u)}} w_{(u,ex)} = 1 \Big)$$

By Theorem 16, $\mathrm{LFP}(P^{\otimes})$, is the only vector in $\mathbb{R}^n$ for which $\varphi_5(\mathbf{w})$ holds true. In other words, φ_5 uniquely identifies $\mathrm{LFP}(P^{\otimes})$. Now, associate a variable $z_{(v,T),(v',T')}$ with each probability $p'_{(v,T),(v',T')}$, where $v = (b, en)$ and $v' = (b, ex)$, and define: $\varphi_6(\mathbf{y}, \mathbf{w}, \mathbf{z}) \equiv \bigwedge_{((v,T),(v',T')) \in \mathrm{Case3}} (z_{(v,T),(v',T')} y_v = w_{((en,T),(ex,T'))} y_{v'})$.

Observe that $\bigwedge_{j=1}^{6} \varphi_j$ has a unique solution, and the values of variables $\mathbf{z}$ in this solution identify the probabilities p' on transitions of $M'_{A,B}$. By the qualitative methods of section 4, we compute the underlying graph $H'_{A,B}$ of $M'_{A,B}$, and we compute the SCCs of $H'_{A,B}$ that contain either an accepting node or an accepting edge.

Let us define a revised finite Markov chain, $M''_{A,B}$, in which we remove all bottom SCCs in $M'_{A,B}$ that contain an accepting node or edge, and replace them by a new absorbing node v^*, with a probability 1 transition to itself. Transitions that were directed into these accepting bottom SCCs are now directed to v^*. Furthermore, in $M''_{A,B}$ we also remove all nodes that can not reach v^*, and all transitions into those nodes. (Technically, some nodes of $M''_{A,B}$ may no longer have full probability on the transitions leaving them, but that is ok for our purposes.)

Now, recall from standard Markov chain theory (see, e.g., [Bil95]) that for such a finite (sub)Markov chain $M''_{A,B}$, there is a *linear* system of equations $\mathbf{t} = F(\mathbf{t})$, over variables t_{u,v^*}, where u is any node of $M''_{A,B}$, and where the coefficients in the linear system $F(\mathbf{t})$ are the probabilities p' on transitions of $M''_{A,B}$, such that the least fixed point solution, $\mathrm{LFP}(F)$, of $\mathbf{t} = F(\mathbf{t})$ assigns to variable t_{u,v^*} the probability that v^* is reachable from u. (In particular, one of the linear equations is $t_{v^*,v^*} = 1$.) Moreover, because we have eliminated from $M''_{A,B}$ all nodes that can not reach v^*, $\mathrm{LFP}(F)$ is the *unique* solution to this linear system. Thus consider the formula: $\varphi_7(\mathbf{w}, \mathbf{t}) \equiv (\mathbf{t} = F(\mathbf{t}))$. Thus the quantifier-free formula $\bigwedge_{j=1}^{7} \varphi_j$ has a unique solution in the reals, and the values assigned to variables $t_{(u,v^*)}$ in this solution identify the probability of reaching an accepting SCC from node u in $M'_{A,B}$.

For initial node $u^* = (v_0, \{q_0\})$ of $M'_{A,B}$, and rational $p \in [0,1]$, the following ***ExTh***$(\mathbb{R})$ sentence, ψ, is true in $\mathbb{R}$ iff $P_A(L(B)) \geq p$:
$\psi \equiv \exists \mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \mathbf{t} \ \bigwedge_{j=1}^{7} \varphi_j \ \wedge (\ t_{u^*,v^*} \geq p)$. ∎

Better complexity bounds can be obtained for the class of linear RMCs and for bounded RMCs.

Theorem 20 *For a linear RMC A and Büchi automaton B, the probability $P_A(L(B))$ is rational and can be computed exactly in polynomial time in $|A|$, and exponential time in $|B|$. If B is deterministic then the time is polynomial in both $|A|$ and $|B|$.*

Proof. Use subset construction on B to construct the deterministic automaton B', and take the product with A to obtain the RMC $A \otimes B'$. If the given RMC A is linear, then the product RMC $A \otimes B'$ is also a linear RMC and obviously can be constructed in time polynomial in $|A|$ and $|B'|$. As shown in (the full version of) [EY05a], the exit probabilities of a linear RMC are rational and can be computed in time polynomial in the size of the RMC. Applying that algorithm on $A \otimes B'$ we can compute explicitly the conditioned summary Markov chain of $A \otimes B'$, which is $M'_{A,B}$, including the exact transition probabilities, in time polynomial in $|A|, |B'|$. We can identify the accepting bottom SCCs with the same complexity, and then solve a linear system to compute the probability that a trajectory of $M'_{A,B}$ starting at the initial state $u^* = (v_0, \{q_0\})$ hits an accepting bottom SCC. ■

For bounded RMCs we can achieve polynomial time if the size of the Büchi automaton is bounded (though the time bound is very impractical).

Theorem 21 *For a fixed Büchi automaton B, given a bounded RMC, A, and a rational value $p \in [0, 1]$, we can decide whether $P_A(L(B)) \geq p$ in time polynomial in $|A|$.*

Proof. If the Büchi automaton B is fixed, then the deterministic automaton B' has bounded size. Taking the product with a bounded RMC A results in another bounded RMC $A \otimes B'$ (note that the number of entries and exits of A gets multiplied by the number of states of B'). The termination probabilities of a bounded RMC are in general irrational, but, as shown in [EY05a], we can answer in polynomial time comparison questions concerning them, using a procedure for the existential theory of the reals with a bounded number of variables.

We summarize below the method from [EY05a]. First the bounded RMC ($A \otimes B'$ in this case) is preprocessed to identify and remove the vertex-exit pairs with 0 probability. Now use variables $x_{(en,ex)}$ only for the set D of entry-exit pairs (en, ex) of the components of $A \otimes B'$ that have nonzero probability; note that there is a bounded number d of such pairs. Let x' be the restriction of the variable vector x of vertex-exit probabilities to these variables $x_{(en,ex)}$ for $(en, ex) \in D$. Then the exit probabilities for all the vertex-exit pairs (u, ex) can be expressed as rational functions of these entry-exit variables. Specifically, for every vertex-exit pair (u, ex) (including the entry-exit pairs) we can construct in polynomial time two polynomials $f_{(u,ex)}(x'), g_{(u,ex)}(x')$ such that $q^*_{(u,ex)} = f_{(u,ex)}(q'^*)/g_{(u,ex)}(q'^*)$, where q'^* is the restriction of the vector q^* to the set

D of (nonzero) entry-exit pairs. The polynomials $f_{(u,ex)}(x'), g_{(u,ex)}(x')$ have rational coefficients of polynomial bit size, and have total degree at most n, the number of vertices. As shown in [EY05a], the vector q'^* is the (unique) minimal nonzero solution to the following set $C(x')$ of constraints: $f_{(en,ex)}(x') = g_{(en,ex)}(x') \cdot x_{(en,ex)}$ and $x_{(en,ex)} > 0$ for all entry-exit pairs $(en, ex) \in D$, and $\sum_{ex} x_{(en,ex)} \leq 1$ for all entries en of each component of the RMC. This solution q'^* of $C(x')$ can be extended to compute the vector q^* for all vertex-exit pairs (u, ex) using the equations $q^*_{(u,ex)} = f_{(u,ex)}(q'^*)/g_{(u,ex)}(q'^*)$. Furthermore the constraint set $C'(x)$ has the property that if we take any other solution r' of $C(x')$ and extend it similarly to all vertex-exit pairs, it results in a vector r that is a fixed point of the original set $x = P(x)$ and hence is $r \geq q^*$. We can therefore determine whether $q^*_{(u,ex)} \leq c$ for some vertex exit pair (u, ex) and rational c by adding to the constraint set $C(x')$ the variable $x_{(u,ex)}$ and constraints $f_{(u,ex)}(x') = g_{(u,ex)}(x') \cdot x_{(u,ex)}$, and $x_{(u,ex)} \leq c$, and invoking an algorithm for the existential theory of the reals with a bounded number of variables. Similarly, we can determine if a vertex u is deficient in polynomial time by adding to $C(x')$ variables $x_{u,ex}$ for all exits $ex \in Ex_i$ of the component of u and adding constraints $f_{(u,ex)}(x') = g_{(u,ex)}(x') \cdot x_{(u,ex)}$ for all $ex \in Ex_i$, and the constraint $\sum_{ex \in Ex_i} x_{(u,ex)} < 1$.

Construct now the Markov chain $M'_{A,B}$, which is the conditioned summary chain of the RMC $A \otimes B'$. We know its set of states, which are the deficient states of the RMC $A \otimes B'$, and its transitions. We do not compute explicitly the values of the transition probabilities, which are irrational numbers, but rather compute them symbolically as rational functions of the vector x' of the entry-exit probabilities of the RMC $A \otimes B'$. Namely, note that the non-exit probability $\mathrm{ne}(u)$ of a vertex u of $A \otimes B'$ is $\mathrm{ne}(u) = 1 - \sum_{ex \in Ex_i} f_{(u,ex)}(x')/g_{(u,ex)}(x')$. The polynomials $f_{(u,ex)}(x'), g_{(u,ex)}(x')$ have total degree n, so $\mathrm{ne}(u)$ is a rational function $f_u(x')/g_u(x')$ where f_u, g_u are polynomials of total degree $\leq dn = O(n)$, also with rational coefficients of polynomial bit-size, and f_u, g_u can be easily constructed in polynomial time. It follows from the definition of the conditioned summary chain $M'_{A,B}$ that the transition probabilities are also rational functions of x' that can be constructed in polynomial time.

We determine the accepting states and accepting edges, and thus the accepting bottom SCCs of the chain $M'_{A,B}$. As in the proof of Theorem 19, we define a revised Markov chain $M''_{A,B}$ by removing all accepting bottom SCCs and replacing them with a new absorbing node v^*; all transitions going to accepting bottom SCCs are directed now to v^*. The desired probability $P_A(L(B))$ is equal to the probability that a trajectory of $M''_{A,B}$ starting at the initial state $u^* = (v_0, \{q_0\})$ hits v^*. If we had the transition probabilities explicitly, we would compute this probability $P_A(L(B))$ by setting up and solving a linear system of equations. By Cramer's rule, $P_A(L(B))$ is equal to the ratio of the determinants of two matrices, $det(F)/det(G)$, whose entries are the transition probabilities, and the constants 0,1. Now the transition probabilities are represented symbolically by rational functions in x', so the probability $P_A(L(B))$ is equal to the ratio $det(F(x'))/det(G(x'))$ of the determinants of two matri-

ces $F(x')$, $G(x')$ whose entries are ratios of polynomials of total degree $O(n)$. Clearing the denominators in the matrix $F(x')$, we can write it as $F(x') = F_1(x')/f_2(x')$ where $f_2(x')$ is the product of all the denominators (a polynomial of total degree $O(n^3)$) and $F_1(x')$ is a matrix whose entries are polynomials of total degree at most $O(n^3)$. Since x' has a fixed number d of variables, each of these polynomials has at most $O(n^{3d})$ terms and can be computed explicitly in polynomial time. We have $det(F(x')) = det(F_1(x'))/(f_2(x))^n$. The numerator $det(F_1(x'))$ is a polynomial $f_1(x')$ of total degree at most $O(n^4)$ and has at most $O(n^{4d})$ terms. As in [EY05a] we can compute $f_1(x')$ explicitly using interpolation, by substituting a sufficient number of tuples for the variables (e.g., $O(n^4)$ values for each variable) and solving a linear system of equations to compute the coefficients of all the possible $O(n^{4d})$ terms of $f_1(x')$. The denominator $(f_2(x))^n$ is also a polynomial of total degree $O(n^4)$ and can be computed easily. Similarly $det(G(x'))$ can be computed as the ratio $g_1(x')/g_2(x')$ of two polynomials, and hence $P_A(L(B)) = f_1(x')g_2(x')/f_2(x')g_1(x') = f(x')/g(x')$ is expressed as the ratio of two polynomials $f(x'), g(x')$ of total degree $O(n^4)$.

We can then test whether $P_A(L(B)) \geq p$ by invoking a procedure for the existential theory of the reals with bounded number of variables on the set of constraints consisting of the system $C(x')$ for the RMC $A \otimes B'$ defined above, constraints $(f_u(x'))^2 > 0$ for all deficient vertices u of the RMC $A \otimes B'$ (recall $\text{ne}(u) = f_u(x')/g_u(x')$, thus $(f_u(x'))^2 > 0$ iff $\text{ne}(u) \neq 0$), $t \cdot g(x') = f(x')$ where t is a new variable that stands for $P_A(L(B))$, and $t \geq p$. The constraints $C(x')$ and $(f_u(x'))^2 > 0$ for deficient vertices u ensure that there is a unique solution which is q'^*, the vector of entry-exit probabilities of $A \otimes B'$, and the constraints $t \cdot g(x') = f(x')$, $t \geq p$ imply that $P_A(L(B)) \geq p$.

■

7 Qualitative Model Checking for Linear Temporal Logic

We build on both the techniques developed in the previous sections for model checking of RMCs with respect to automata specifications, as well as the techniques developed for LTL model checking of flat Markov Chains in [CY95]. The algorithm of [CY95] for model checking LTL properties of flat Markov chains employs an iterative approach, whereby the chain is refined in each iteration and the formula is simplified by elimination of temporal operators, until at the end the formula becomes propositional and can be verified directly. There are serious technical obstacles however for effectively extending this approach to the recursive setting, and this is not what we do. Instead, we follow a different approach which is more global in nature. We use an idea from another method of [CY95], used there for another purpose (for 'Extended Temporal Logic'), and we extend it with other techniques to handle recursion and LTL.

We are given RMC A and an LTL formula φ. We assume wlog that the RMC starts at the entry node en_{init} of component A_0 of A, which has no exit. First, we construct

from A (the graph of) the summary Markov chain M'_A; we only need the nodes and edges of M'_A and not the precise transition probabilities. We identify the formula φ with its parse tree T. The leaves of the tree are labelled with atomic propositions and its non-leaf nodes are labelled with temporal or Boolean connectives. Let n be the number of propositions and internal nodes of T; number the propositions and internal nodes from 1 to n bottom-up: first the propositions and then the internal nodes. For each i, let φ_i be the subformula of φ corresponding to the tree T_i rooted at node i.

Let M_A be the (infinite) Markov chain represented by the RMC A. Let $X = x_0x_1x_2\ldots$ be an infinite trajectory of M_A starting at some state $x_0 = \langle \beta, u\rangle$. We define the *type* of the trajectory to be a Boolean vector t of length n, where for each i, $t_i = 1$ iff X satisfies the formula φ_i. From the definition of the satisfaction of LTL formulas it follows that the pair (u, t) satisfies the following properties:

1. If φ_i is a proposition p, then $t_i = 1$ if p holds at u, else $t_i = 0$.
2. If i is an internal node of the parse tree labelled with a Boolean connective $\neg$ (resp. $\vee$, $\wedge$) and has child j (resp. children j, l), then $t_i = \neg t_j$ (resp. $t_i = t_j \vee t_l$, $t_i = t_j \wedge t_l$).
3. If i is labelled with a temporal connective $\mathcal{U}$ and has children j, l, i.e., $\varphi_i = \varphi_j \mathcal{U} \varphi_l$, then (a) if $t_l = 1$ then also $t_i = 1$, and (b) if $t_j = t_l = 0$ then also $t_i = 0$.

We call any pair (u, t) consisting of a vertex u of the RMC A and a Boolean n-vector t *consistent* if it satisfies these three properties. Similarly we say that the pair (x_0, t) consisting of a state $x_0 = \langle \beta, u\rangle$ of M_A and a vector t is consistent if the pair (u, t) is consistent. Observe that if (u, t) is consistent then the temporal coordinates of t (those corresponding to nodes of φ labelled with a temporal connective) determine uniquely the rest of the coordinates of t because of properties (1), (2).

Consider a trajectory $X = x_0x_1x_2\ldots$ and suppose that we know the type s of its suffix $x_1x_2\ldots$. Then we can determine uniquely the type t of X from s and the state x_0 (more precisely, the vertex u of x_0) as follows: The coordinates t_i corresponding to propositions are determined from u by property (1). For the internal nodes of the parse tree, proceed bottom-up in the tree. Let i be an internal node and suppose that we have determined the coordinates corresponding to the children of i. If i is labelled by a Boolean connective then t_i is determined by property (2) of consistency. If i is labelled by a temporal connective then t_i is determined by property (3) unless i is labelled (i) $\bigcirc$ (Next) or (ii) it is labelled $\mathcal{U}$ (Until) with children j, l and $t_j = 1, t_l = 0$. In case (i), if i has child j, i.e. $\varphi_i = \bigcirc\varphi_j$ then $t_i = s_j$; in case (ii) we must have $t_i = s_i$. Thus, these two properties (i), (ii) and the consistency conditions (1-3) above determine uniquely t from u and s. We will say that t is the type *backwards implied* for the vertex u and the state x_0 from type s.

The backward implication extends to finite paths: If $\pi = x_0x_1\ldots x_k$ is a finite path of M_A and s is a type consistent with the final state x_k, then there is a unique type t that is backwards implied from s and π for the initial state x_0 of the path and its vertex.

We construct a graph G as follows. The nodes of G are all pairs (u,t) where u is a node of the summary chain M'_A and t is a Boolean n-vector such that the pair (u,t) is consistent. We include an edge $(u,t) \to (v,s)$ between two nodes of G if M'_A has an edge $u \to v$ and (a) either the edge is not a summary edge and t is the type that is backwards implied from s for the node u, or (b) $u \to v$ is a summary edge, i.e. $u = (b, en)$, $v = (b, ex)$ for some box b, and there is a path π in the RMC corresponding to the summary edge (i.e., a path π in M_A from $\langle \epsilon, u \rangle$ to $\langle \epsilon, v \rangle$) such that t is the type that is backwards implied from π and s.

We can check in case (b) whether there exists a path π in the RMC from u to v satisfying the above requirement, as follows: Construct a Recursive State Machine (RSM) $\hat{A}$, called the *augmented RSM*, which has a component $\hat{A}_i$ for each component A_i of the RMC A. There is a node (u,t) for each vertex u of A and each type t that is consistent with u; if u is an entry or exit of a component A_i, then (u,t) is an entry or exit of the corresponding component $\hat{A}_i$. If b is a box of A_i mapped to A_j, then there is a corresponding box $\hat{b}$ in $\hat{A}_i$ that is mapped to $\hat{A}_j$; for every entry en of A_j and consistent tuple t, the box $\hat{b}$ has a corresponding call port which we will denote $((\hat{b}, en), t)$ (the vertex is labelled with the same propositions as en), and we define similarly the return ports of $\hat{b}$. Note that the vertices of the form (u,t), where $u = (b, en)$ or $u = (b, ex)$ was a call port or return port of box b of A, are now ordinary nodes of $\hat{A}$. We include an edge $(u,t) \to (v,s)$ for each pair of vertices $(u,t), (v,s)$ of $\hat{A}$ such that t is the type backwards implied from s for u, and either A contains an edge $u \to v$, or $u = (b, en)$ and $v = (\hat{b}, en)$ for some box b of A, or $u = (\hat{b}, ex)$ and $v = (b, ex)$. (Note: The reason that we introduced new call ports and return ports is that the trajectories of the Markov chain M_A contain explicit steps corresponding to the recursive calls and returns from the calls. This is a small technical detail.) It is easy to see now that there is a path π in the RMC A from $u = (b, en)$ to $v = (b, ex)$ (with empty context) that corresponds to the summary edge $u \to v$ and such that t is the type that is backwards implied from π and s iff the RSM $\hat{A}$ contains a path from (u,t) to (v,s) with empty context (i.e., $M_{\hat{A}}$ has a path from $\langle \epsilon, (u,t) \rangle$ to $\langle \epsilon, (v,s) \rangle$). We can check this by applying the RSM reachability algorithm of [ABE+05] to the augmented RSM $\hat{A}$.

Consider again a trajectory $X = x_0 x_1 x_2 \ldots$ of M_A. For each j, let t^j be the type of the path $x_j x_{j+1} \ldots$. By our previous remarks, the pair (x_j, t^j) is consistent. Also, note that t^j is the type backwards implied by t^{j+1} and x_i. Let $\hat{X}$ be the sequence $(x_0, t^0), (x_1, t^1), (x_2, t^2) \ldots$; we call this the *augmented trajectory* corresponding to X. It corresponds to a trajectory of the RSM $\hat{A}$.

Recall that there is a mapping ρ from trajectories X of the original Markov chain M_A to a trajectory of the summary chain M'_A, or to the symbol $\star$, with the property that $P_A(\rho^{-1}(\star)) = 0$. Suppose that $\rho(X) \neq \star$. Then $\rho(X)$ consists of the vertex parts $u_0 u_{i1} u_{i2} \ldots$ of a subsequence $x_0 x_{i1} x_{i2} \ldots$ of X obtained by shortcutting subpaths of X by summary edges. The mapping ρ can be extended to the augmented trajectory $\hat{X}$: $\rho(\hat{X}) = (u_0, t^0), (u_{i1}, t^{i1}) \ldots$ is obtained from the corresponding subsequence of $\hat{X}$ by

keeping only the vertex parts and the types. By our construction of the graph G, $\rho(\hat{X})$ is a path of G.

If $(v_0, s_0), (v_1, s_1), (v_2, s_2) \ldots$ is a sequence of vertex-type pairs, then the *projection* of the sequence on the first component is the sequence $v_0, v_1, v_2 \ldots$ of vertices.

Lemma 22 *1. Every finite or infinite path of G projected on the first component yields a path of M'_A.*
2. Conversely, every path of M'_A is the projection of at least one path in G.

Proof. (1) follows directly from the construction of G. (2) is obvious for finite paths by construction. For infinite paths, note that every path of M'_A is the image $\rho(X)$ of some trajectory X of M_A. Let $\hat{X}$ be the augmented trajectory. Then $\rho(\hat{X})$ is a path of G whose projection is $\rho(X)$. ■

Recall that a vertex u of A is included in summary chain M'_A iff $ne(u) > 0$. Call a pair (u, t) *probable* if there is positive probability that a trajectory of A starting at u does not exit the component of u (does not terminate) and has type t. We denote by $P'(u, t)$ the probability that a trajectory from u has type t conditioned on the event that it does not exit u's component.

Lemma 23 *1. If G contains an edge $(u, t) \to (v, s)$ and (v, s) is probable then (u, t) is also probable.*
2. In particular, in every strongly connected component C of G, either all nodes are probable or none is.

Proof. With nonzero probability, a trajectory starting at u will go to v following the edge $u \to v$ (if it is an ordinary edge or a nesting edge) or following some path π (if $u \to v$ is a summary edge) such that t is the type implied back by s and π. There is positive probability that the trajectory from v does not exit v's component and has type s. If this happens, then the trajectory from u will also not exit its component and will have type t. This proves claim 1. Claim 2 follows immediately from 1. ■

Let H be the subgraph of G consisting of probable nodes. By the above lemma, in order to compute H, it suffices to identify which strongly connected components of G are the bottom SCCs of H. Then H consists of all the nodes that are ancestors of these bottom SCCs. Once we compute the graph H, we can answer the qualitative model checking problem: The trajectories of the given RMC A satisfy the given formula φ almost surely if and only if H does not include any node of the form (en_{init}, t), where en_{init} is the initial node of A (the entry node of the top component) and t is a type with $t_n = 0$. Note that n corresponds to the root of the parse tree of φ, i.e., $\varphi_n = \varphi$, so (en_{init}, t) probable with $t_n = 0$ would mean that there is positive probability that a trajectory starting at en_{init} does not satisfy φ. (Recall that the top component has no exit, so all the trajectories from en_{init} do not exit its component.)

A trajectory X of the RMC (i.e. of the infinite chain M_A) maps with probability 1 to a trajectory $X' = \rho(X)$ of the summary chain M'_A, and the augmented trajectory $\hat{X}$ maps to an augmented trajectory $\hat{X}' = \rho(\hat{X})$ that is a path in G. Call a trajectory X of M_A *typical* if $X' = \rho(X)$ is defined and all pairs of $\hat{X}' = \rho(\hat{X})$ are probable, i.e. if $\hat{X}'$ is a path of the subgraph H. It follows easily from the Markov property that the set of typical trajectories of the RMC starting at the initial state has probability 1. More generally it is easy to show the following:

Lemma 24 *For every vertex u of the RMC A with $ne(u) > 0$, the probability that a trajectory starting at u does not exit its component and is typical with type t is $ne(u)P'(u,t)$.*

We wish to find the improbable nodes of G and remove them to obtain H. As we noted, it suffices to identify the bottom SCCs of H. From the definition of G, if G contains a path from (u,t) to (v,s) then M'_A contains a path from u to v. Therefore, for every SCC C of G, the first components of all the nodes of C belong to the same SCC K of M'_A. We will say that the SCC C *corresponds* to K.

Lemma 25 *If C is a bottom SCC of H, then it corresponds to a bottom SCC K of M'_A.*

Proof. Let (u,t) be a node of C. A trajectory X of the RMC starting at u does not exit u's component with probability $ne(u)$, and conditioned on this event, with probability 1 it is typical and its summary image $\rho(X)$ is absorbed in a bottom SCC of the summary chain M'_A. Since (u,t) is probable, the summary image $\rho(X)$ of such a typical trajectory of type t must be the projection of a path in H starting at (u,t). Since C is a bottom SCC of H, it follows that its corresponding SCC K of M'_A must be also a bottom SCC. ■

We will now give a necessary condition for a node of G to be probable. Consider a summary edge $(u,t) \to (v,s)$ of G. We say that the edge is probable if the nodes are probable. We label the edge with a subset of $\{1, \ldots n\}$ as follows. A label $l \in \{1, \ldots n\}$ is included in the subset iff the infinite chain $M_{\hat{A}}$ of the augmented RMC $\hat{A}$ has a path from $\langle \epsilon, (u,t) \rangle$ to $\langle \epsilon, (v,s) \rangle$ that goes through some node $\langle \beta, (z,r) \rangle$ with $r_l = 1$. This can be determined in polynomial time in the size of $\hat{A}$ using the algorithm for Recursive State Machines of [ABE+05].

Lemma 26 *If (u,t) is probable, then it satisfies the following condition. For every node i of (the parse tree of) φ labelled $\mathcal{U}$, with corresponding subexpression $\varphi_i = \varphi_j \mathcal{U} \varphi_l$, if $t_i = 1$ then node (u,t) can reach in H (and in G) some probable node (v,s) with $s_l = 1$ or some probable summary edge whose label set includes l.*

Proof. Consider a typical trajectory $X = \langle \epsilon, u \rangle x_1 x_2 \ldots$ starting at u that does not exit its component and has type t. Its summary image $Y = \rho(X) = u v_{i1} v_{i2} \ldots$,

consists of the vertex parts of a subsequence $\langle \epsilon, u \rangle x_{i1} x_{i2}$ of X. Some suffix $x_k x_{k+1} \ldots$ of X satisfies φ_l. Since X is typical, its augmented trajectory $\hat{X}$ maps to a path $\hat{Y} = \rho(\hat{X}) = (u,t)(v_{i1}, t^{i1}) \ldots$ in H. If v_k is included in the summary path Y, then the node (v_k, t^k) is in the path $\hat{Y}$ of H, hence it is a probable node with $t^k_l = 1$. If v_k is not included in the summary path Y, then let v_p, v_r be the nodes that are included with p the maximum index less than k and r the minimum index greater than k. Then $(v_p, t^p), (v_r, t^r)$ is a probable summary edge with label l. ■

It is convenient for the purposes of the analysis to refine the summary graph M'_A into a multigraph M''_A as follows. For each summary edge $u = (b, en) \to v = (b, ex)$ consider all paths of the RMC that give rise to the edge, i.e. paths of the form $\langle \epsilon, u \rangle \to \langle b, en \rangle \to \ldots \langle b, ex \rangle \to \langle \epsilon, v \rangle$. For every type s for the final state, each path implies backwards a type t for u. Let us call two paths *equivalent* if they induce the same mapping from types s at v to types t at u. This gives us a partition of the paths into equivalence classes. Replace the summary edge $u \to v$ with a set of parallel edges, one for each equivalence class. Do the same for all summary edges of M'_A and let M''_A be the resulting multigraph. We can view M''_A also as a (refined) Markov chain where the probability of the summary edges is divided among the parallel edges that replaced it according to the total probability of all paths in each equivalence class. (We do not actually perform this transformation; it is only for the purposes of the analysis.) The multigraph M''_A has the property that for every edge $u \to v$ (whether an ordinary, a summary, or a nesting edge) and every type s for v there is a unique type t that is implied for u by s and the edge. Note that, by construction, the graph G contains an edge $(u,t) \to (v,s)$; we will say that the edge $u \to v$ of M''_A is a projection of the edge $(u,t) \to (v,s)$ of G. (More than one parallel summary edges of M''_A from u to v may be the projection of the same edge of G.) We can extend the notion of projection to paths of G. Obviously M'_A and M''_A have the same SCCs (replacing an edge by a set of parallel edges does not change the SCCs).

Lemma 27 *Let C be a SCC of G and let K be the corresponding SCC of M''_A. The following are equivalent.*

1. *For every node (v,s) of C, every edge $u \to v$ of K is a projection of some edge $(u,t) \to (v,s)$ of C into (v,s).*
2. *Every finite path in K is a projection of some path in C.*
3. *No other SCC of G corresponding to K is ancestor of C.*

The proof is nontrivial but it is very similar to the proof of Lemma 5.10 of [CY95], so we will omit it and refer to that paper.

The characterization of bottom SCCs of H is given by the following Theorem.

Theorem 28 *A SCC C of G is a bottom SCC of H if and only if the following three conditions are satisfied.*

1. *C corresponds to a bottom SCC K of M'_A.*
2. *No other SCC of G corresponding to K is ancestor of C.*
3. *For every subexpression $\varphi_i = \varphi_j \mathcal{U} \varphi_l$ of φ, either all nodes (u,t) of C have $t_i = 0$ or there is a node $(v,s) \in C$ with $s_l = 1$ or there is a summary edge of C whose label set includes l.*

Proof. Suppose that C is a bottom SCC of H. Then C satisfies conditions 1 and 3 by Lemmas 25 and 26 respectively. Suppose that it does not satisfy (2). Then from Lemma 27 there is a finite path β of K that is not the projection of any path in C. Let (u,t) be any node of C. A trajectory of M''_A starting at u contains with probability 1 the path β (in fact the path occurs infinitely often in the trajectory). Such a trajectory is not the projection of any path in C. It follows that (u,t) is not probable.

Conversely, suppose C satisfies the three conditions. We show that C contains all probable pairs (u,t) whose first component u is in K. From this it follows that C is the only SCC of H that corresponds to K, and C is a bottom SCC of H because any descendant SCC must then also correspond to K. To prove the above fact we show the following lemma. The converse of the theorem follows once we prove the lemma. ■

Lemma 29 *Suppose that C satisfies the three conditions of Theorem 28. For every probable pair (u,t) with $u \in K$, the following are true for each $i = 1, \ldots, n$.*

1. *There is a node (u,t') of C such that t and t' agree in the first i coordinates.*
2. *There is a finite path $\alpha(u,t,i)$ of M''_A starting at u such that the type of almost all trajectories of the RMC from u that do not exit u's component, whose summary image has prefix $\alpha(u,t,i)$, agrees with t in the first i coordinates.*

Proof. We use induction on i. The basis, $i = 1$ is trivial: φ_1 is a proposition and part (1) is satisfied by any node (u,t') of C with first component u. Note that C has such a node since every path of K is the projection of a path of C (by condition (2) and Lemma 27). As for part (2), we let $\alpha(u,t,1)$ be the trivial path that consists of node u.

For the induction step, the lemma follows trivially if φ_i is a proposition, or node i of the parse tree of φ is labelled with a Boolean connective, or if it is labelled with $\mathcal{U}$ and the value of t_i is determined uniquely by property (3) of consistency, i.e., $\varphi_i = \varphi_j \mathcal{U} \varphi_l$ and $t_l = 1$ or $t_i = t_j = 0$. In these cases, if we have a probable pair (u,t) and a node (u,t') of C such that t and t' agree in the first $i-1$ coordinates, then they must agree also in the ith coordinate. Also, we may let $\alpha(u,t,i) = \alpha(u,t,i-1)$. There are two remaining cases.

Case 1: i is labelled with the next operator. Suppose that $\varphi_i = \bigcirc\varphi_j$. Let (u,t) be a probable pair and take any typical trajectory X of the RMC starting at u that does not exit u's component and has type t. Consider the summary image $\rho(X)$ of X, let v be the second node of $\rho(X)$ and s the type of the suffix of X from (this occurrence of) v on. Since $u \in K$, K is a bottom SCC of M'_A, and there is an edge $u \to v$, it follows that also $v \in K$.

Subcase 1.1. Suppose first that u is not a call port. Then v is simply the second vertex of the trajectory X. Clearly, v is in the same component of the RMC as u, the trajectory does not exit v's component and since it is typical, the pair (v,s) is probable. By the induction hypothesis, there is a node (v,s') of C such that s and s' agree in the first $i-1$ coordinates. By condition (2) of the theorem and Lemma 27, (v,s') has an incoming edge from a node (u,t') of C with first component u. This node (u,t') fulfils the required property 1: the first i coordinates of t' are determined from the first $i-1$ coordinates of s' in the same way that the corresponding coordinates of t are determined from s, and note that $t_i = s_j$ and $t'_i = s'_j$, hence $t_i = t'_i$. For part 2, we let $\alpha(u,t,i)$ be $u \to v$ followed by $\alpha(v,s,i-1)$.

Subcase 1.2. Suppose that u is a call port $u = (b, en)$. The second node v of $\rho(X)$ is either the entry en of the component of A corresponding to the box b, or it is a return port $v = (b, ex)$ of the box. In the first case, the argument is exactly the same as above; note that the suffix of X from $v = en$ on does not exit v's component and (v,s) is a probable pair. So suppose that $v = (b, ex)$ is a return port of the box b, and let π be the prefix of X from u to v. The type t at u is the type that is backwards implied by the path π and the type s. Again, (v,s) is a probable pair and thus C contains a node (v,s') where s' agrees in the first $i-1$ coordinates with s. The equivalence class of the path π corresponds to one of the parallel summary edges of M''_A, say edge a, from u to v. From Lemma 27 it follows that C contains a corresponding edge $(u,t') \to (v,s')$, such that t' is the type that is backwards implied from the path π and s'. Since s and s' agree in the first $i-1$ coordinates, the same is true for all the types implied at corresponding nodes of the path π, and thus also at u, the first node of the path, as well as at the second node of the path π. Since t_i and t'_i are equal to the respective coordinates l at the second node, it follows that t and t' agree in the first i coordinates. As for part 2, we let $\alpha(u,t,i)$ be the summary edge a from u to v (corresponding to the path π) followed by the path $\alpha(v,s,i-1)$.

Case 2: Node i is labelled with the Until operator. Suppose that $\varphi_i = \varphi_j \mathcal{U} \varphi_l$, and that $t_j = 1, t_l = 0$ (we took care of the other possibilities for t). Take a typical trajectory X of the RMC starting at u that does not exit u's component and has type t. Let $X = \langle \epsilon, u \rangle x_1 x_2 \ldots$, and let $Y = \rho(X) = u y_1 y_2 ...$ be its summary image. We will distinguish cases according to the value of t_i.

Subcase 2.1: $t_i = 1$. Let m be the smallest index such that the suffix $x_m x_{m+1} \ldots$ of X satisfies φ_l; such an index m exists by the definition of $\mathcal{U}$, and for all $k < m$, the corresponding suffix from x_k on satisfies φ_j. Suppose first that the summary image

$Y = \rho(X)$ includes the node corresponding to x_m, i.e. $x_m = \langle \beta, v \rangle$ and all subsequent $x_q, q > m$ include the context β. Let $s = t^m$ be the type of the suffix of X from x_m on. Since the trajectory is typical, (v, s) is a probable pair, and the summary chain contains a path π' from u to v (namely, the summary image of the prefix of X up to x_m). Therefore, v is in the same bottom SCC K as u. By the induction hypothesis, C contains a node (v, s') such that s' agrees with s in the first $i-1$ coordinates. From Lemma 27, the path π' from u to v in K is the projection of a path in C from some node (u, t') to (v, s'). It follows then that t and t' agree in the first i coordinates (they agree on coordinate i because all nodes (z, q) along the path have $q_j = 1$ and the final node has $s'_l = s_l = 1$). We let the path $\alpha(u, t, i)$ be π' followed by the path $\alpha(v, s, i-1)$.

Suppose that the image trajectory $Y = \rho(X)$ in the summary chain does not include the node corresponding to x_m, i.e. it is shortcut by a summary edge (w, v), where $w = (b, en)$, $v = (b, ex)$ for some box b. That is, for some indices $p < m$, $q > m$, we have $x_p = \langle \beta, w \rangle$, $x_q = \langle \beta, v \rangle$ and all states of the trajectory X between x_p and x_q include the context βb. Let $r = t^p$, $s = t^q$. Again $v \in K$ and the pair (v, s) is probable. By the induction hypothesis, C contains a node (v, s') such that s that agrees with s' in the first $i-1$ coordinates. From Lemma 27, the SCC C contains a path from some node (u, t') to (v, s') with projection the path π' of M''_A from u to v corresponding to the prefix of X up to x_q. If we consider this prefix of X up to x_q, substitute s' for the type at x_q in place of s, and then infer backwards the types t'^k at the preceding states $x_k, k < q$, obviously all the types t'^k will agree in the first $i-1$ coordinates with t^k. This implies in particular that the type at x_m will have the lth coordinate $t'^m_l = 1$. Since the jth coordinate in all the preceding states is 1, it follows that $t'_i = 1$, hence t' agrees with t in the first i coordinates. We let again the path $\alpha(u, t, i)$ be π' followed by the path $\alpha(v, s, i-1)$.

Subcase 2.2: $t_i = 0$. Recall that $t_j = 1, t_l = 0$. We consider two further subcases.

Subcase 2.2.1: There is a typical trajectory $X = \langle \epsilon, u \rangle x_1 x_2 \ldots$, starting at u that does not exit u's component, has type t, and some suffix of X from some state x_m on satisfies $\varphi_j = \varphi_l = 0$. The arguments are very similar to the case $t_i = 1$. Consider the summary image $Y = \rho(X)$. Either it contains the node corresponding to x_m or the node is shortcut by a summary edge. Consider the second case; the first case is similar and simpler. For some indices $p < m$, $q > m$, we have $x_p = \langle \beta, u \rangle$, $x_q = \langle \beta, v \rangle$ and all states of the trajectory X between x_p and x_q include the context βb. Let $r = t^p$, $s = t^q$. Again $v \in K$ and the pair (v, s) is probable, so by the induction hypothesis, there is a node $(v, s') \in C$ such that s' agrees with s in the first $i-1$. There is a path in C from some node (u, t') to (v, s') with projection the path π' of K from u to v that is the summary image of the prefix of X up to x_q. Again we can infer backwards the types and conclude that t, t' agree in the first i coordinates.

Subcase 2.2.2: For every typical trajectory X, starting at u that does not exit u's component and has type t, every suffix of X satisfies $\varphi_j = 1$ or $\varphi_l = 1$. Consider such a typical trajectory $X = \langle \epsilon, u \rangle x_1 x_2 \ldots$. Suppose that there is an index m such that the

suffix $x_m....$ satisfies $\varphi_l = 1$, and let m be the smallest such index. Since $\varphi_l = 0$ for smaller indices $k < m$, it follows that $\varphi_j = 1$ for them, hence from the semantics of the Until operator it follows that trajectory X satisfies φ_i, contradicting the assumption that $t_i = 0$. Therefore, it must be the case that every suffix $x_m....$ satisfies $\varphi_l = 0$ and hence $\varphi_j = 1$. We will argue that for any $v \in K$, every probable pair (v, s) has $s_l = 0, s_j = 1$, and there is no edge $w \to v$ of K that is the projection of a probable summary edge into (v, s) with label l.

Let (v, s) be a probable pair with $v \in K$ and consider the finite path $\alpha(v, s, i-1)$. Every trajectory of the summary chain M''_A starting at u will contain this path as a subpath with probability 1. In other words, for almost every trajectory X of the RMC that starts at u and does not exit u's component, its summary image $\rho(X)$ will contain this path. Since the type of almost all trajectories whose ρ image has prefix $\alpha(v, s, i-1)$ agrees with s in the first $i-1$ coordinates, and since every suffix of X satisfies $\varphi_l = 0$ and $\varphi_j = 1$, it follows that $s_l = 0$ and $s_j = 1$.

Suppose that there is a probable summary edge $(w, r) \to (v, s)$ whose label includes l, and with projection the edge $a = w \to v$ of K. Let π be a $u - v$ path of the RMC corresponding to the summary edge. We know that $r_l = s_l = 0$ and $r_j = s_j = 1$. Consider the path consisting of the summary edge a followed by the path $\alpha(v, s, i-1)$. Every trajectory of the summary chain M''_A starting at u will contain this path as a subpath with probability 1. Thus, almost every non-exiting trajectory X of the RMC starting at u will have an image $\rho(X)$ that contains this path. Let $X = \langle \epsilon, u \rangle x_1 x_2 \ldots$ be such a typical trajectory of type t where x_p gets mapped to w in the summary chain, x_q is mapped to v, and the subpath $\pi = x_p \ldots x_q$ is mapped to the summary edge $a = w \to v$. We may assume wlog (it happens a.s.) that the type of the suffix from x_q on agrees with s in the first $i-1$ coordinates. If we infer the types along the path π backwards from x_q, some intermediate state x_m of the path will have $t^m_l = 1$ because the summary edge includes label l, and clearly this label depends only on the first $i-1$ coordinates of s. By our assumption, no suffix of the trajectory satisfies $\varphi_j = \varphi_l = 0$. It follows that the whole trajectory satisfies $\varphi_i = \varphi_l \mathcal{U} \varphi_j$, contradicting our assumption that $t_i = 0$. We conclude that there is no probable summary edge $(w, r) \to (v, s)$ in G with label l where $w, v \in K$.

In the same way that we showed that if one node of a SCC of G is probable then all the nodes are probable, we can argue that the same property is true if we restrict attention to the first $i-1$ coordinates of the types. This implies that for all nodes (v, s') of C we have $s'_l = 0$ and $s'_j = 1$. Also, no summary edge $(w, r') \to (v, s')$ is labelled l. (Since $l \leq i-1$, if there was a $w - u$ path π that yielded such a l-labelled summary edge, then the above argument would still apply by restricting types to the first $i-1$ coordinates). By condition (3) of Theorem 28, it follows that all nodes (v, s') of C have their ith coordinate $s'_i = 0$. So we may let (u, t') be the node of C that agrees with (u, t) in the first $i-1$ coordinates. We may take $\alpha(u, t, i) = \alpha(u, t, i-1)$. ∎

Summarizing, the qualitative model checking algorithm for a RMC A and a LTL

formula φ works as follows.

1. Construct the graph of the summary chain M'_A.
2. Generate all consistent pairs (u, t), $u \in M'_A$, t a type.
3. Construct the graph G on the consistent pairs.
4. Find the strongly connected components of G, and construct the DAG of SCCs.
5. While there is a bottom SCC that violates one of the conditions of Theorem 28, remove it from G.
6. If the final graph H contains a node (en_{init}, t) with $t_n = 0$ then reject, else accept.

By our analysis, the final graph is the subgraph H of G induced by the probable pairs.

Step 1 (which depends only on the RMC A, not on the formula φ) can be done in polynomial space in A [EY05a]. The rest of the steps can be done in polynomial time in the size of the graph G and the RSM $\hat{A}$, both of which are polynomial in $|A|$ and exponential in $|\varphi|$ (more specifically, the exponent only depends on the number of temporal operators in φ). If A is a 1-exit RMC, or bounded RMC, or linear RMC, then Step 1 can be done in polynomial time in A. Thus:

Theorem 30 *Given RMC A and LTL formula φ, we can check whether A satisfies φ with probability 1 in PSPACE in A and EXPTIME in φ. If A is a 1-exit RMC or a bounded RMC or linear RMC then the time complexity is polynomial in A.*

Conversely, we can show that qualitative model checking of LTL formulas requires exponential time.

Theorem 31 *The qualitative problem of determining whether a given RMC A satisfies a LTL formula φ with probability 1 (i.e., whether $P_A(\varphi) = 1$) is EXPTIME-hard (thus EXPTIME-complete). Furthermore, this holds even if the RMC is fixed and each component has one entry and one exit.*

The proof is similar to the proof of Theorem 15 for automata, and to the proof of a theorem in [BEM97] showing that LTL model checking for (non-probabilistic) Pushdown Systems (equivalent to RSMs) is EXPTIME-hard. The latter proof encodes a finite accepting computation tree of an alternating linear space Turing Machine as a finite path in a RSM, and uses LTL formulas to check that the path is consistent with an accepting sequence of configurations of the alternating Turing machine. Since all finite paths have non-zero probability in an RMC, we can in principle use the same result and ignore probabilities on transitions to get EXPTIME-hardness for RMCs. (The proof of [BEM97] is only sketched in their online Tech. Report and leaves several details out.)

Further, as our construction shows, the RSM itself (and in our setting the RMC) can be of fixed size, with each component having 1 entry and 1 exit. We omit the detailed proof.

8 Quantitative Model Checking of LTL Properties

We are given a Recursive Markov Chain A and an a LTL formula φ. We are also given a rational number p, and we want to determine whether the probability $P_A(\varphi)$ that a trajectory of A satisfies φ is at least (or at most) p. As mentioned in Section 2, the probability $P_A(\varphi)$ is in general irrational and thus it cannot be computed explicitly. We will construct a system of polynomial equations and inequalities in a set of real variables, one of which stands for the desired probability $P_A(\varphi)$. The system will be constructed in such a way that it has a unique solution. Then we will attach the inequality $P_A(\varphi) \geq p$ (or $P_A(\varphi) \leq p$) and invoke a procedure for the existential theory of the reals to check whether the resulting system is satisfiable.

First we set up the system (1a) $\mathbf{x} = P(\mathbf{x})$ of fixed point equations for the RMC A which contains one variable $x_{(u,ex)}$ for every vertex u and exit ex of u's component. Recall that we can compute in PSPACE in $|A|$ the set $Q' = \{u \in Q \mid ne(u) > 0\}$ of deficient vertices. We add to (1a) the constraints (1b) $\mathbf{x} \geq 0$; (1c) $y_u = 1 - \sum_{ex \in Ex_{c(u)}} x_{(u,ex)}$ for every vertex u; (1d) $y_u > 0$ for every vertex u in Q'; and (1e) $y_u = 0$ for every vertex u in $Q - Q'$. Let (1) be the system of constraints (1a)-(1e). From the Unique Fixed Point Theorem for RMCs, Theorem 16, system (1) has a unique solution $(\mathbf{x}, \mathbf{y})$, and this solution is $x_{(u,ex)} = q^*_{(u,ex)}$ and $y_u = ne(u)$.

Now, we carry out the algorithm for the qualitative model checking. As a result we compute all probable pairs (u, t). For a deficient vertex u and a type t, let $P'(u, t)$ be the probability that a trajectory X starting at u has type t conditioned on the event that X does not exit u's component. We have a corresponding variable $z(u, t)$ (we only need to include the probable pairs, since the others have probability 0). These variables satisfy several constraints:
(2a) $\sum_t z(u, t) = 1$ for all $u \in Q'$.
(2b) If u is not a call port, then $z(u, t) = \sum_{(v,s)} p'_{u,v} z(v, s)$, where $p'_{u,v}$ is the probability of transition $u \to v$ in the summary Markov chain M'_A, and the sum ranges over all probable pairs (v, s) such that H contains an edge $(u, t) \to (v, s)$.
(2c) If u is a call port, $u = (b, en)$, then $z(u, t) = p'_{u,en} \sum_s z(en, s) + \sum_{(v,s)} p'_{u,v} f_{u,v,t,s} z(v, s)$, where the first sum ranges over all types s such that H contains an edge $(u, t) \to (en, s)$, and the second sum ranges over all exits $v = (b, ex)$ of the box b and types s such that H contains an edge $(u, t) \to (v, s)$ and $f_{u,v,t,s}$ is the fraction of the probability of $u - v$ paths of the RMC for which the type s at v implies backwards the type t at u.

These constraints are justified by the following lemma.

Lemma 32 *Probabilities $P'(u, t)$ satisfy constraints 2a-2c.*

Proof. (2a) is obvious: Every trajectory that starts at u and does not exit must have some type, and the types t for which (u,t) is not probable (for which we did not include variables) have probability 0.

For (2b), consider the typical trajectories X that start at u and do not exit u's component. Then $Y = \rho(X)$ is a trajectory of M'_A. With probability $p'_{u,v}$ the second vertex is v, the trajectory does not exit the component of v (which is the same as that of u), and the trajectory from v on has type s with probability $P'(v,s)$; the type of X will be t iff there is an edge $(u,t) \to (v,s)$ in H.

For (2c), consider again the typical trajectories X that start at $u = (b, en)$ and do not exit u's component, and let $Y = \rho(X)$. There are two kinds of such trajectories. The first kind consists of those that never exit the box b, that is, they enter the component corresponding to b at the entry node en and never reach an exit. This happens with probability $p'_{u,en}$. The subsequent trajectory from en does not exit its component, and has type s will probability $P'(en,s)$; the type of the whole trajectory X will be t iff there is an edge $(u,t) \to (en,s)$ in H. The second kind of trajectories X consists of those that eventually exit the box b at some return port $v = (b, ex)$, (i.e. v is the second node of the image trajectory $Y = \rho(X)$ in M'_A), but then the rest of X from v does not reach the exit of the component of v (which is the same as the component of u). This happens with probability $p'_{u,v}$. The rest of the trajectory from v has type s with probability $P'(v,s)$. Then X has type t if the $u - v$ path that was followed to exit the box b implies back t from s; this happens with probability $p'_{u,v} f_{u,v,t,s}$. ∎

The transition probabilities $p'_{u,v}$ of M'_A are rational functions of the probabilities captured by the variables $(\mathbf{x}, \mathbf{y})$ of system (1). The quantities $f_{u,v,t,s}$ are in general irrational, so we cannot compute them explicitly; however, we will later present a system of constraints with a unique solution that gives precisely these quantities. Suppose for now that we have also determined the parameters $f_{u,v,t,s}$. Then the constraints (2) form a linear system in the variables $z(u,t)$. It turns out that this system has a unique solution.

Lemma 33 *The system (2) of linear equations in the variables $z(u,t)$ has a unique solution.*

Proof. From the summary chain M'_A we form a refined chain M''_A as described in the previous section, where we replace every summary edge $u \to v$ of M'_A by a set of parallel edges, one for each equivalence class of $u - v$ paths, and we distribute the transition probability of the edge $u \to v$ among these parallel edges proportionately to the probability of the paths of the RMC that they represent. Then $p'_{u,v} f_{u,v,t,s}$ is the sum of transition probabilities on the parallel edges of M''_A corresponding to the classes where s at v maps back to t at u.

Let us also introduce parallel edges and edge weights in the graph H: Replace every summary edge $(u,t) \to (v,s)$ of H by a set of parallel edges, one for each equivalence

class of $u-v$ paths that imply back t at u from s at v. Let H' be the resulting multigraph. Now every edge a' of H' corresponds to a unique edge a of M''_A; give weight to edge a' equal to the transition probability on edge a of M''_A. The edge weights of H' do not make H' into a Markov chain because weights out of a node may not sum to 1. Note that every path of H' corresponds to (we'll say, *projects onto*) a unique path of M''_A. Furthermore, for every node (v, s) of H' and every edge $a = u \to v$ of M''_A, the graph H' contains a unique corresponding edge a' into (v, s); the head of the edge is a node (u, t) for some t.

The proof of the lemma uses a similar technique to that of Proposition 5.11 in [CY95]. Write the system of equations (2b-2c) as $\mathbf{z} = B\mathbf{z}$ where $\mathbf{z}$ is the vector of variables $z(u, t)$ and B is the coefficient matrix of the right-hand side. The rows and columns of B are indexed by the probable pairs, and the entry $B[(u, t), (v, s)]$ is equal to the sum of the weights of the edges $(u, t) \to (v, s)$ of H'. If α is a finite path (sequence of edges) of M''_A or H', then we denote by $w(\alpha)$ the product of the probabilities (or weights) of the edges along the path α and call it the weight of α. Consider the jth power B^j of B. Then $B^j[(u, t), (v, s)]$ is the sum of the weights of the paths α' of length j of H' from (u, t) to (v, s). Every such path projects to a unique path α of M''_A from u to v, and α has the same weight.

A trajectory of the (refined) summary Markov chain M''_A starting at any node u hits with probability 1 eventually a bottom SCC K. Recall from Lemma 29 that if v is any node of K and s any type such that (v, s) is probable, then there is a finite path $\alpha(v, s, n)$ such that any trajectory of M''_A from v with prefix $\alpha(v, s, n)$ has type s with probability 1. A trajectory from u that hits K will eventually with probability 1 contain the path $\alpha(v, s, n)$ as a subpath. If β is finite a path of M''_A from a node u that hits a bottom SCC K and includes a subpath $\alpha(v, s, n)$ for some $v \in K$ and type s such that (v, s) is probable, then we will say that β is *determined*. We assign to such a β a unique type t, which is the type that is backwards implied by the prefix from u to the occurrence of v right before the subpath $\alpha(v, s, n)$ and the type s at v. Clearly, H' contains a path corresponding to β starting at (u, t) (the path goes on to (v, s) and continues from there). Furthermore, H' has no path corresponding to β starting at any other node (u, t') for any other type $t' \neq t$. The reason is that such a path would have to go to a node (v, s') with $s' \neq s$ followed by a path corresponding to $\alpha(v, s, n)$; but then (v, s') cannot be a probable pair, because almost all trajectories of M''_A from v with prefix $\alpha(v, s, n)$ have type s.

Let $d_j(u, t, v)$ be the sum of the weights (probabilities) of the paths β of M''_A of length j from u to v that are determined of type t. Let $d_j(u, t) = \sum_v d_j(u, t, v)$, let $d_j(u) = \sum_t d_j(u, t)$, and let $\epsilon_j(u) = 1 - d_j(u)$. The last quantity $\epsilon_j(u)$ is the probability that a path of M''_A of length j starting at u is not determined. Thus, by the definition and our discussion above, $\epsilon_j(u) \to 0$ as $j \to \infty$.

Consider a path β from u to v of length j that is determined of type t, i.e. β contributes weight $w(\beta)$ to $d_j(u, t, v)$. As we said above, no node (u, t') with $t' \neq t$ has

a path corresponding to β. For every node (v,s) of H' there is a path ending at (v,s) that corresponds to β; this path has to start at (u,t). Therefore β contributes weight $w(\beta)$ to $B^j[(u,t),(v,s)]$ for every s, and does not contribute to any $B^j[(u,t'),(v,s)$ with $t' \neq t$. Therefore, for any s we have $d_j(u,t,v) \leq B^j[(u,t),(v,s)]$.

Conversely, consider a path β of M''_A that contributes its weight to $B^j[(u,t),(v,s)]$, i.e. β is the projection of a path in H' of length j from (u,t) to (v,s). If β is determined then its type must be t and its weight is included in $d_j(u,t,v)$. The set of paths of length j that are not determined have total weight $\epsilon_j(u)$. Therefore, $B^j[(u,t),(v,s)] \leq d_j(u,t,v) + \epsilon_j(u)$. Since $\lim_{j\to\infty} \epsilon_j(u) = 0$, it follows that $\lim_{j\to\infty}(B^j[(u,t),(v,s)] - d_j(u,t,v)) = 0$.

Note that if a path β is determined then so are all its extensions and they have the same type t. Therefore, $d_j(u,t)$ is a non-decreasing function of j, and since it is bounded from above by 1, it has a limit $d_\infty(u,t)$. If $\mathbf{z}$ is any solution to the system (2) then for any j it satisfies $\mathbf{z} = B^j\mathbf{z}$. Thus, $z(u,t) = \sum_{(v,s)} B^j[(u,t),(v,s)]z(v,s)$ $= \sum_{(v,s)}(B^j[(u,t),(v,s)] - d_j(u,t,v))z(v,s) + \sum_{(v,s)} d_j(u,t,v)z(v,s)$. As j tends to ∞, the first term tends to 0 and the second term tends to $d_\infty(u,t)$. It follows that $z(u,t) = d_\infty(u,t)$. ∎

We will now construct a system of constraints that determines uniquely the parameters $f_{u,v,t,s}$. Recall the augmented Recursive State machine $\hat{A}$ that we constructed. We add weights to its edges and convert it to a weighted RSM; it will not necessarily be a RMC because the weights out of a node may not sum to 1. The edges of $\hat{A}$ are of the form $(u,t) \to (v,s)$. If A contains the edge $u \to v$ then we let the weight of $(u,t) \to (v,s)$ be the probability of the edge $u \to v$. The other cases are that $u = (b, en)$ and $v = (\hat{b}, en)$, or $u = (\hat{b}, ex)$ and $v = (b, ex)$; in these cases we give these edges weight 1.

Let $u = (b, en)$, $v = (b, ex)$ be a call port and a return port of a box b, and let π be a path in the RMC corresponding to the summary edge $u \to v$ in the summary graph, i.e. π is a path $\langle \epsilon, u\rangle \to \langle b, en\rangle \to \ldots \langle b, ex\rangle \to \langle \epsilon, v\rangle$, where all the intermediate nodes include b in the context. For every type s for the final vertex v, we can infer uniquely types for the vertices along the path, and in particular a type t for the initial vertex u. Thus, the augmented RSM $\hat{A}$ contains for every type s a unique path $\hat{\pi}_s$ corresponding to π which goes from a vertex (u,t) for some t (with empty context) to (v,s) and that path $\hat{\pi}_s$ has the same weight as the probability of the path π. The path $\hat{\pi}_s$ is composed of an edge from (u,t) to an entry $((\hat{b}, en), t')$ of the box $\hat{b}$, then a path that eventually reaches an exit $((\hat{b}, ex), s')$ of the box $\hat{b}$ and finally an edge from the exit to (v,s). Suppose that we have at hand for each entry (en, t') and exit (ex, s') of each component $\hat{A}_i$ of the weighted RSM $\hat{A}$ the sum $h(en, t', ex, s')$ of the weights of all the paths from the entry to the exit. Then we can use them to compute the quantity $x_{en,ex} \cdot f_{u,v,t,s}$ which is the sum of the probabilities of all the paths π corresponding to summary edges $u \to v$ for which type s at v is mapped back to type t at u. Namely,

(3a) $x_{en,ex} \cdot f_{u,v,t,s} = \sum h(en, s', ex, t')$ where the summation ranges over all s', t' such that $\hat{A}$ has edges $(u,t) \to ((\hat{b}, en), t')$ and $((\hat{b}, ex), s') \to (v, s)$.

We introduce a variable $h(u, t, ex, s)$ for every pair consisting of a vertex (u, t) of $\hat{A}$ and an exit (ex, s) of its component, to represent the sum of the weights of all the paths from (u, t) that exit at (ex, s). We will construct a set of fixed point equations, whose solution will be the desired weights. The fixed point equations are similar to the system of equations for an RMC, given in Section 2. The only difference now is that the weights on the edges out of a vertex may not sum to 1. Let (3b) $\mathbf{h} = \hat{P}(\mathbf{h})$ be this system of equations. We add the constraints (3c): $\mathbf{h} \geq 0$. Finally we add the following constraints (3d): $\sum_t h(u, t, ex, s) = x(u, ex)$ for every triple u, ex, s where u is a vertex of component A_i of the RMC A, ex is an exit of the same component and s is a type. Note that (u, t) is a vertex of component $\hat{A}_i$ and (ex, s) is an exit of the component. The justification for these constraints is the following. For every path π from u to ex (with empty context) and every type s there is a unique corresponding path in $\hat{A}$ to (ex, s), and this path starts at a vertex (u, t) for some t and has weight equal to the probability of the path π. Summing over all such paths π gives the constraint (3d).

We claim now that having fixed the x variables (from constraints (1)), the system (3b-d) has a unique solution. First, note that the intended solution $\mathbf{h}$ representing the weights of the vertex-exit paths is the least fixed point solution of the system (3b-c). This can be shown in the same way as it is shown for Recursive Markov Chains. Namely, if we start with $\mathbf{h} = 0$ and apply repeatedly the operator $\hat{P}$ then the vector will converge to the least fixed point solution and this coincides with the desired vector of weights. If we pick a fixed point solution that is strictly greater in some component $h(u, t, ex, s)$ than the correct weights, then the solution will violate a constraint (3d). We conclude that the system (3b-d) has a unique solution. It follows then that (3a) determine uniquely the parameters $f_{u,v,t,s}$.

To summarize, we have three sets of constraints (1),(2),(3). The quantities $p'_{u,v}$ in constraints (2) (the transition probabilities of the summary chain) are ratios, so we first rewrite (2) to clear the denominators so that they become also polynomial equations. If we want to check whether the probability $P_A(\varphi)$, that a trajectory of A satisfies φ, is at least a given threshold p, then we add the constraint (4) $\sum z(en_{init}, t) \geq p$, where the summation ranges over all t with $t_n = 1$. Then we call a procedure for the existential theory of the reals on the system (1-4). Similarly we can determine if the probability is less than p. We can also approximate the probability $P_A(\varphi)$ within any number k of bits of precision by doing a binary search using the above procedure k times.

The size of the system of constraints is polynomial in $|A|$ and exponential in $|\varphi|$. It follows that the complexity is polynomial space in $|A|$ and exponential in $|\varphi|$. For linear RMCs, we can solve the constraints explicitly by solving a series of linear systems of equations.

Theorem 34 *Given RMC A, LTL formula φ and rational value p, we can determine whether the probability $P_A(\varphi)$ that a trajectory of A satisfies φ is $\geq$ (or $\leq$) p in space*

polynomial in A and exponential in φ. If A is a linear RMC, then we can compute $P_A(\varphi)$ exactly in time polynomial in A and exponential in φ.

9 Conclusions

We presented algorithms and lower bounds for the model checking of Recursive Markov chains against ω-regular specifications, given by Büchi automata or LTL formulas. The complexity results for the two formalisms turn out to be similar, though they require different algorithms because of the difference of the two formalisms in expressiveness and succinctness. We studied both the qualitative problem, i.e., testing whether the specification is satisfied with probability 1 or 0, and the quantitative problem, i.e. determining whether the probability of satisfaction meets a given threshold, or approximating the probability to a desired precision. For a given RMC A and property (Büchi automaton B or LTL formula φ) we showed that the qualitative problem can be solved in PSPACE in the size of the RMC and EXPTIME in the size of the property, and on the other hand it is EXPTIME-complete even for fixed RMC A. We saw that the bottleneck with respect to the RMC is the computation of the deficient (survivor) vertices u of the RMC, i.e., the vertices that have positive probability $\mathrm{ne}(u) > 0$ of not terminating. We showed that once we identify these vertices, then the rest of the qualitative model checking problem involves an intricate combinatorial analysis which depends polynomially on the size of the RMC. As a consequence, for several important classes of RMCs (linear, bounded, and 1-exit RMCs) the complexity is polynomial in the size of the RMC. Also if the property is given by a deterministic Büchi automaton B, then the complexity in $|B|$ is polynomial. For the quantitative problem we showed that it can be solved in PSPACE in the size of the RMC and EXPSPACE in the size of the property.

In the non-recursive case, there has been algorithmic work on the model checking of systems that have both probabilistic and non-probabilistic actions, modeled by a Markov Decision Process (or equivalently a Concurrent Markov Chain) (see e.g., [CY95, Var85]) resulting in algorithms and tight complexity results. In the recursive case, this is in general not possible: as shown in [EY05b], there are ω-regular properties whose model checking problem already for Recursive Markov Decision Processes (even for 1-exit linear RMDPs) is undecidable.

Acknowledgement: Research partially supported by NSF Grants CCF-04-30946 and CCF-07-28736.

References

[ABE+05] R. Alur, M. Benedikt, K. Etessami, P. Godefroid, T. Reps, and M. Yannakakis. Analysis of recursive state machines. *ACM Trans. Program. Lang. Syst.*, 27(4):786–818, 2005.

[ABKPM06] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. B. Miltersen. On the complexity of numerical analysis. In *21st IEEE Computational Complexity Conference*, 2006.

[AEY01] R. Alur, K. Etessami, and M. Yannakakis. Analysis of recursive state machines. In *Proc. of 13th Int. Conf. on Computer-Aided Verification*, pages 304–313, 2001.

[BEM97] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Applications to model checking. In *CONCUR'97*, pages 135–150, 1997.

[BGR01] M. Benedikt, P. Godefroid, and T. Reps. Model checking of unrestricted hierarchical state machines. In *Proc. of ICALP'01*, volume 2076 of *LNCS*, pages 652–666, 2001.

[Bil95] P. Billingsley. *Probability and Measure*. J. Wiley and Sons, 3rd edition, 1995.

[BKS05] T. Brázdil, A. Kučera, and O. Stražovský. Decidability of temporal properties of probabilistic pushdown automata. In *Proc. of STACS'05*, 2005.

[BPR96] S. Basu, R. Pollack, and M. F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *J. of the ACM*, 43(6):1002–1045, 1996.

[Can88] J. Canny. Some algebraic and geometric computations in PSPACE. In *Prof. of 20th ACM STOC*, pages 460–467, 1988.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[DEKM99] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic models of Proteins and Nucleic Acids*. Cambridge U. Press, 1999.

[EKM04] J. Esparza, A. Kučera, R. Mayr. Model checking probabilistic pushdown automata. In *LICS 2004*, 2004.

[EKM06] J. Esparza, A. Kucera, R. Mayr. Model cheking probabilistic pushdown automata. *Logical Methods in Computer Science*, 2(1), pp. 1-31, 2006.

[EY05a] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of non-linear equations. In *Proc. of 22nd STACS'05*. Springer, 2005. Full expanded version available from http://homepages.inf.ed.ac.uk/kousha/.

[EY05b] K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *Proc. of 32nd Int. Coll. on Automata, Languages, and Programming (ICALP'05)*, 2005.

[EY05c] K. Etessami and M. Yannakakis. Algorithmic Verification of Recursive Probabilistic State Machines. In *Proc. 11th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'05)*, 2005.

[FKK+] R. Fagin, A. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with "back buttons". *ACM Symp. on Theory of Computing*, pages 484–493, 2000. Full version in *Annals of Applied Probability*, 11, pp 810-862, 2001.

[GGJ76] M. R. Garey, R. L. Graham, and D. S. Johnson. Some NP-complete geometric problems. In *8th ACM Symp. on Theory of Computing*, pages 10–22, 1976.

[HJV05] P. Haccou, P. Jagers, and V. A. Vatutin. *Branching Processes: Variation, Growth, and Extinction of Populations*. Cambridge U. Press, 2005.

[Har63] T. E. Harris. *The Theory of Branching Processes*. Springer-Verlag, 1963.

[KA02] M. Kimmel and D. E. Axelrod. *Branching processes in biology*. Springer, 2002.

[Kwi03] M. Kwiatkowska. Model checking for probability and time: From theory to practice. *Proc. 18th IEEE LICS*, pages 351-360, 2003.

[MS99] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[Pnu77] A. Pnueli. The temporal logic of programs. In *Proc. 18th Symp. on Foundations of Comp. Sci.*, pages 46–57, 1977.

[PZ93] A. Pnueli and L. D. Zuck. Probabilistic verification. *Information and Computation*, pages 1-29, 1993.

[Ren92] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals. parts i,ii, iii. *J. of Symbolic Computation*, pages 255–352, 1992.

[Tiw92] P. Tiwari. A problem that is easier to solve on the unit-cost algebraic ram. *Journal of Complexity*, pages 393–397, 1992.

[Var85] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. of 26th IEEE Symp. on Foundations of Comp. Sci.*, pages 327–338, 1985.

[VW86] M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proc. 1st Symp. on Logic in Comp. Sci. (LICS)*, pages 322–331, 1986.

[YE05] M. Yannakakis and K. Etessami. Checking LTL Properties of Recursive Markov Chains. In *Proc. 2nd Intl. Conf. on Quantitative Evaluation of Systems*, IEEE, 2005.