Recursive Markov Decision Processes and Recursive Stochastic Games

Kousha Etessami[1] and Mihalis Yannakakis[2]

[1] LFCS, School of Informatics, University of Edinburgh
[2] Department of Computer Science, Columbia University

Abstract. We introduce Recursive Markov Decision Processes (RMDPs) and Recursive Simple Stochastic Games (RSSGs), and study the decidability and complexity of algorithms for their analysis and verification. These models extend Recursive Markov Chains (RMCs), introduced in [EY05a,EY05b] as a natural model for verification of probabilistic procedural programs and related systems involving both recursion and probabilistic behavior. RMCs define a class of denumerable Markov chains with a rich theory generalizing that of stochastic context-free grammars and multi-type branching processes, and they are also intimately related to probabilistic pushdown systems. RMDPs & RSSGs extend RMCs with one controller or two adversarial players, respectively. Such extensions are useful for modeling nondeterministic and concurrent behavior, as well as modeling a system's interactions with an environment.
We provide upper and lower bounds for deciding, given an RMDP (or RSSG) A and probability p, whether player 1 has a strategy to force termination at a desired exit with probability at least p. We also address "qualitative" termination, where $p = 1$, and model checking questions.

1 Introduction

Markov Decision Processes (MDPs) are a fundamental formalism for modeling control optimization problems in sequential stochastic environments. They have found widespread applications in many fields (see, e.g., [Put94,FS02]). They have also been studied extensively in recent years for verification of probabilistic systems. Stochastic games generalize MDPs with multiple players, and in their 2-player zero-sum version are also known as Competitive MDPs (see [FV97]). Simple Stochastic Games (SSGs) [Con92] are a special class of 2-player zero-sum stochastic games, where the goal of one player is to reach a given terminal state, while the other aims to avoid it. SSGs generalize parity games and other important games for model checking, and the termination problem for finite SSGs already presents a well-known algorithmic challenge: it is in NP $\cap$ coNP, but no P-time algorithm is known ([Con92]).

Recursive Markov Chains (RMCs) were introduced and studied in our earlier work ([EY05a,EY05b]) as a natural model of probabilistic procedural programs and systems exhibiting both recursion and probabilistic behavior. There we provided strong upper and lower bounds for both reachability and ω-regular model checking questions for RMCs. Informally, a RMC consists of a (finite) collection of finite state Markov chains that can call each other in a potentially recursive manner. RMCs define a class of denumerable Markov chains with a rich theory generalizing that of Stochastic Context-Free Grammars (SCFGs) (see, e.g., [MS99]) and Multi-Type Branching Processes ([Har63]), both of which correspond to 1-exit RMCs: RMCs in which each component Markov chain has 1 terminating exit state where it can return control back to a component that called it. RMCs are also intimately related to probabilistic Pushdown Systems (pPDSs), which have also been studied recently in connection to verification of probabilistic programs ([EKM04,BKS05]).

For verification, it is natural and useful to extend RMCs with nondeterministic choice, where some states are controlled by the system while others exhibit probabilistic behavior. Indeed, finite MDPs have been studied extensively for verification of probabilistic systems, and optimized verification tools already exist for them (see, e.g.,[CY98,Var85,dAKN$^+$00,Kwi03]; [Kwi03] is a recent survey). SSGs extend MDPs with a second (adversarial) player. Like non-probabilistic game graphs, they can also be used to model and analyze the interactions between a controlled (but probabilistic) system and an (adversarial) environment.

In this paper we focus on precisely such extensions of RMCs: we introduce *Recursive Markov Decision Processes* (RMDPs) and *Recursive Simple Stochastic Games* (RSSGs), which define natural classes of countable MDPs and SSGs, respectively. In the stochastic dynamic programming literature, MDPs are studied under many different reward criteria, such as average reward, discounted reward, etc. Our focus here is on verification of probabilistic systems, and for this purpose we study RMDPs and RSSGs under reachability criteria which are central to any analysis like model checking. In particular, we ask the *quantitative termination* question: given an RMDP (or RSSG) A and a probability p, is there a strategy for the controller where (regardless of the strategy used by the adversary, in the case of RSSGs) the process terminates at a desired exit with probability at least p (or at most p)? We also ask the *qualitative* question of whether the controller has a strategy to force termination with probability 1. Lastly, we address model checking questions.

Our positive results apply primarily to 1-exit RMDPs and 1-exit RSSGs, which correspond to controlled and game extensions, respectively, of both SCFGs and Multi-Type Branching Processes (MT-BPs). Branching processes are an important class of stochastic processes, dating back to the early work of Galton and Watson in the 19th century (they studied the single-type case, a subcase of 1-exit 1-entry 1-component RMCs), and continuing in the 20th century in the work of Kolmogorov, Sevastianov, Harris and others for MT-BPs and beyond (see, e.g., [Har63]). These have been used to model a wide variety of applications, including in population genetics ([Jag75]), nuclear chain reactions, and RNA modeling in computational biology (based on SCFGs) ([SBH$^+$94]). SCFGs are also fundamental models in statistical natural language processing (see, e.g., [MS99]). 1-exit RMDPs correspond to a controlled version of MT-BPs (and SCFGs): the reproduction of some types can be controlled, while the dynamics of other types is probabilistic as in ordinary MT-BPs. This model would also be suitable for analysis of population dynamics under worst-case (or best-case) assumptions for some types and probabilistic assumptions for others. Such controlled MT-BPs can be readily translated to 1-entry, 1-exit RMDPs, where the number of components is bounded by the number of types (a reverse translation is possible, but will not in general preserve the number of components, i.e., 1-entry, 1-exit RMDPS with a bounded number of components are more general than MT-BPs with a bounded number of types). Thus, our results on 1-exit RMDPs apply, among other things, to such controlled MT-BPs; these do not appear to have been studied in the rich Branching Process literature. Indeed, even some basic algorithmic problems about SCFGs and MT-BPs had received limited attention prior to our work in [EY05a,EY05b].

We now outline our main results in this paper:

- We show that the Least Fixed Point solution of certain systems of nonlinear min/max equations captures optimal termination probabilities for 1-exit RMDPs & 1-exit RSSGs. These equations generalize linear Bellman's equations for finite MDPs (see, e.g., [Put94,FV97]) and also generalize the monotone systems of nonlinear equations for RMCs that we studied in ([EY05a]).
- We show a quite nontrivial *Stackless & Memoryless (S&M) Determinacy* result for 1-exit RSSG termination, whereas we observe this fails badly even for 2-exit RMDPs (namely, optimal strategies of any kind do not always exist for 2-exit RMDP termination; one must make do with ϵ-optimal strategies).
- Using the equations, we show that quantitative termination for 1-exit RMDPs and 1-exit RSSGs is decidable in PSPACE. This matches our PSPACE upper bound for the special case of 1-exit RMCs in [EY05a] and, as shown there, it can not be improved without resolving a long standing open problem in the complexity of numerical computation, namely the square-root sum problem.
- Using S&M-determinacy, we show qualitative termination (where $p = 1$) can be decided in NP for 1-exit RMDPs , and in $\Sigma_2^P \cap \Pi_2^P$ for 1-exit RSSGs.
- For the special case of *linearly recursive* 1-exit RMDPs (RSSGs), we show that the exact optimal, and rational, termination probabilities can be computed in polynomial time (in NP∩co-NP, respectively).
- Lastly, and unfortunately, we show that for multi-exit RMDPs *&* RSSGs the situation is far worse: even qualitative termination for general RMDPs is undecidable, even when the number of exits in bounded by a fixed constant and the RMDP is restricted to be linearly-recursive. It is even undecidable, for any fixed $\epsilon > 0$, to distinguish whether the optimal value is 1 or $< \epsilon$. So optimal probabilities can not be approximated in a strong sense, with any resources. Furthermore, we show undecidability applies already to qualitative model checking of 1-exit RMDPs, against regular or LTL properties. Our undecidability results are derived from classic and recent undecidability results for Probabilistic Finite Automata (PFA) [Paz71,CL89,BC03]. We show PFAs can be viewed as essentially a special case of multi-exit RMDPs.

Related work. Both MDPs and Stochastic Games have a vast literature, dating back to Bellman and Shapley (see, e.g., [Put94,FS02,FV97]). MDPs are studied in both finite state and infinite state variants. Verification of finite state MDPs, also called concurrent Markov chains, has been studied for a long time (see, e.g., [CY98,CY95,Var85,HSP83]). [CY98] provides efficient algorithms for ω-regular model checking of finite MDPs. Model checking tools like PRISM contain optimized implementations of branching-time model checkers for finite MDPs (see, e.g., [dAKN$^+$00,Kwi03]).

Our earlier work [EY05a,EY05b] developed the basic theory of RMCs and studied efficient algorithms for both their reachability analysis and model checking. We showed, among many results, that qualitative model checking of ω-regular properties for 1-exit RMCs can be decided in polynomial time in the size of the RMC, and that quantitative model checking of RMCs can be done in PSPACE in the size of the RMC. As mentioned, 1-exit RMCs correspond to both MT-BPs and SCFGs (see, e.g., [Har63] and [MS99]), while general RMCs are intimately related to probabilistic Pushdown Systems (pPDSs). Model checking questions for pPDSs, for both linear and branching time properties, have also been recently studied in [EKM04,BKS05]. RMDPs and RSSGs are natural extensions of RMCs, introducing nondeterministic and game behavior. Countable state MDPs are studied extensively in the MDP literature (see, e.g., [Put94,FS02]), but the concise representations afforded by RMDPs and its algorithmic properties, appear not to have been studied prior to our work.

2 Basics

A *Recursive Simple Stochastic Game (RSSG)*, A, is a tuple $A = (A_1, \ldots, A_k)$, where each *component graph* $A_i = (N_i, B_i, Y_i, En_i, Ex_i, \mathtt{pl}_i, \delta_i)$ consists of:

- A set N_i of *nodes*. Let $N = \cup_{i=1}^k N_i$ be the (disjoint) union of all nodes of A.
- A distinguished subset of *entry* nodes $En_i \subseteq N_i$, and a disjoint subset of *exit* nodes $Ex_i \subseteq N_i$. Let $En = \cup_{i=1}^k En_i$ and $Ex = \cup_{i=1}^k Ex_i$.
- A set B_i of *boxes*. Let $B = \cup_{i=1}^k B_i$ be the (disjoint) union of all boxes of A.
- A mapping $Y_i : B_i \mapsto \{1, \ldots, k\}$ that assigns to every box (the index of) of a component. Let $Y = \cup_{i=1}^k Y_i$ be the map $Y : B \mapsto \{1, \ldots, k\}$ where $Y|_{B_i} = Y_i$, for $1 \leq i \leq k$.
- To each box $b \in B_i$, we associate a set of *call ports*, $Call_b = \{(b, en) \mid en \in En_{Y(b)}\}$, and a set of *return ports*, $Return_b = \{(b, ex) \mid ex \in Ex_{Y(b)}\}$. Let $Call^i = \cup_{b \in B_i} Call_b$ and let $Call = \cup_{i=1}^k Call^i$ denote all calls in A. Similarly, define $Return^i$ and $Return$.
- We let $Q_i = N_i \cup Call^i \cup Return^i$, denote collectively the nodes, call ports, and return ports, We will use the term *vertex* of A_i to refer to elements of Q_i. We let $Q = \bigcup_{i=1}^k Q_i$ be the set of all vertices of the RSSG A.
- A mapping $\mathtt{pl}_i : Q_i \mapsto \{0, 1, 2\}$ that assigns to every vertex a <u>player</u> (Player 0 represents "chance" or "nature"). We assume $\mathtt{pl}_i(ex) = 0$ for all $ex \in Ex_i$. Let $\mathtt{pl} = \cup_{i=1}^k \mathtt{pl}_i$ denote $\mathtt{pl} : Q \mapsto \{0, 1, 2\}$ where $\mathtt{pl}|_{Q_i} = \mathtt{pl}_i$, for $1 \leq i \leq k$.
- A transition relation $\delta_i \subseteq (Q_i \times (\mathbb{R} \cup \{\bot\}) \times Q_i)$, where for each tuple $(u, x, v) \in \delta_i$, the source $u \in (N_i \setminus Ex_i) \cup Return^i$, the destination $v \in$

$(N_i \setminus En_i) \cup Call^i$, and x is either (i) a real number $p_{u,v} \in [0,1]$ (the transition probability) if $\texttt{pl}(u) = 0$, or (ii) $x = \perp$ if $\texttt{pl}(u) = 1$ or 2. For computational purposes we assume that the given probabilities $p_{u,v}$ are rational. Furthermore they must satisfy the consistency property: for every $u \in \texttt{pl}^{-1}(0)$, $\sum_{\{v' \mid (u, p_{u,v'}, v') \in \delta_i\}} p_{u,v'} = 1$, unless u is a call port or exit node, neither of which have outgoing transitions, in which case by default $\sum_{v'} p_{u,v'} = 0$. Let $\delta = \cup_i \delta_i$ be the set of all transitions of A.

An RSSG A defines a global denumerable Simple Stochastic Game (SSG) $M_A = (V = V_0 \cup V_1 \cup V_2, \Delta, \texttt{pl})$ as follows. The global *states* $V \subseteq B^* \times Q$ of M_A 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 $\langle \beta, u \rangle \in V$ & $(u, x, v) \in \delta$, then $\langle \beta, v \rangle \in V$ and $(\langle \beta, u \rangle, x, \langle \beta, v \rangle) \in \Delta$.
3. if $\langle \beta, (b, en) \rangle \in V$ & $(b, en) \in Call_b$, then $\langle \beta b, en \rangle \in V$ & $(\langle \beta, (b, en) \rangle, 1, \langle \beta b, en \rangle) \in \Delta$.
4. if $\langle \beta b, ex \rangle \in V$ & $(b, ex) \in Return_b$, then $\langle \beta, (b, ex) \rangle \in V$ & $(\langle \beta b, ex \rangle, 1, \langle \beta, (b, ex) \rangle) \in \Delta$.

Item 1 corresponds to the possible initial states, item 2 corresponds to control staying within a component, item 3 is when a new component is entered via a box, item 4 is when control exits a box and returns to the calling component. The mapping $\texttt{pl} : V \mapsto \{0,1,2\}$ is given as follows: $\texttt{pl}(\langle \beta, u \rangle) = \texttt{pl}(u)$ if u is in $Q \setminus (Call \cup Ex)$, and $\texttt{pl}(\langle \beta, u \rangle) = 0$ if $u \in Call \cup Ex$. The set of vertices V is partitioned into V_0, V_1, and V_2, where $V_i = \texttt{pl}^{-1}(i)$.

We consider M_A with various *initial states* of the form $\langle \epsilon, u \rangle$, denoting this by M_A^u. Some states of M_A are *terminating states* and have no outgoing transitions. These are states $\langle \epsilon, ex \rangle$, where ex is an exit node.

An RSSG where $V_2 = \emptyset$ ($V_1 = \emptyset$) is called a maximizing (minimizing, respectively) *Recursive Markov Decision Process* (RMDP); an RSSG where $V_1 \cup V_2 = \emptyset$ is called a *Recursive Markov Chain* (RMC) ([EY05a,EY05b]); an RSSG where $V_0 \cup V_2 = \emptyset$ is called a *Recursive Graph* ([AEY01]); an RSSG where $V_0 = \emptyset$ is called a *Recursive Game Graph* (see [ATM03,Ete04]). We use *1-exit RSSG* to refer to RSSGs where every component has 1 exit. W.l.o.g., we can assume every component has 1 entry, because multi-entry RSSGs can be transformed to equivalent 1-entry RSSGs with polynomial blowup (similar to RSM transformations [AEY01]). This is decidedly not so for exits: 1-exit RSSGs form an important sub-class of RSSGs and are the main focus of our upper bounds. We shall call a RSSG (RMDP, RMC, etc.) *linearly-recursive* (denoted `lr`-RSSG, etc.) if there in no path of transitions in any component from any return port to a call port. `lr`-RMCs are much easier to analyse than general RMCs: reachability probabilities are rational and both reachability analysis and model checking can be performed with the same complexity as for finite Markov chains, using the decomposed Newton's method [EY05a] and techniques we developed in [EY05a,EY05b] (although `lr`-RMCs were not mentioned explicitly in [EY05a,EY05b]).

A basic goal is to answer termination questions for RSSGs: *"Does player 1 have a strategy to force the game to terminate at ex (i.e., reach state $\langle \epsilon, ex \rangle$),* *starting at $\langle\epsilon, u\rangle$, with probability $\geq p$, regardless of how player 2 plays?"*. A *strategy* σ for player i, $i \in \{1,2\}$, is a function $\sigma : V^*V_i \mapsto V$, where, given the history $ws \in V^*V_i$ of play so far, with $s \in V_i$ (i.e., it is player i's turn to play a move), $\sigma(ws) = s'$ determines the next move of player i, where $(s, \perp, s') \in \Delta$. (We could also allow randomized strategies.)

Let Ψ_i denote the set of all strategies for player i. A pair of strategies $\sigma \in \Psi_1$ and $\tau \in \Psi_2$ induce in a straightforward way a Markov chain $M_A^{\sigma,\tau} = (V^*, \Delta')$, whose set of states is the set V^* of histories. Given initial vertex u, a final exit ex in the same component, and a $k \geq 0$, let $q^{k,\sigma,\tau}_{(u,ex)}$ be the probability that, in $M_A^{\sigma,\tau}$, starting at initial state $\langle\epsilon, u\rangle$, we will reach a state $w\langle\epsilon, ex\rangle$ in at most k "steps" (i.e., where $|w| \leq k$). Let $q^{*,\sigma,\tau}_{(u,ex)} = \lim_{k\to\infty} q^{k,\sigma,\tau}_{(u,ex)}$ be the probability of ever terminating at ex, i.e., reaching $\langle\epsilon, ex\rangle$ (the limit exists: the sequence is monotonically non-decreasing & bounded by 1). Let $\mathbf{q}^k_{(u,ex)} = \max_{\sigma\in\Psi_1}\min_{\tau\in\Psi_2} q^{k,\sigma,\tau}_{(u,ex)}$ and let $\mathbf{q}^*_{(u,ex)} = \sup_{\sigma\in\Psi_1}\inf_{\tau\in\Psi_2} q^{*,\sigma,\tau}_{(u,ex)}$. Next, for a strategy $\sigma \in \Psi_1$, let $q^{k,\sigma}_{(u,ex)} = \min_{\tau\in\Psi_2} q^{k,\sigma,\tau}_{(u,ex)}$, and let $q^{*,\sigma}_{(u,ex)} = \inf_{\tau\in\Psi_2} q^{*,\sigma,\tau}_{(u,ex)}$. Lastly, given instead a strategy $\tau \in \Psi_2$, let $q^{k,\cdot,\tau}_{(u,ex)} = \max_{\sigma\in\Psi_1} q^{k,\sigma,\tau}_{(u,ex)}$, and let $q^{*,\cdot,\tau}_{(u,ex)} = \sup_{\sigma\in\Psi_1} q^{*,\sigma,\tau}_{(u,ex)}$.

From very general determinacy results (eg. Martin's "Blackwell determinacy" [Mar98]) it follows that the games M_A are *determined*, meaning that $\sup_{\sigma\in\Psi_1}\inf_{\tau\in\Psi_2} q^{*,\sigma,\tau}_{(u,ex)} = \inf_{\tau\in\Psi_2}\sup_{\sigma\in\Psi_1} q^{*,\sigma,\tau}_{(u,ex)}$. Of course, finite SSGs are even *memorylessly determined* ([Con92]), meaning that the strategies of either player can be restricted to *memoryless strategies* which ignore the history prior to the current position, without harming the optimal outcome. As we shall see, 1-exit RSSGs exhibit memoryless determinacy in an even stronger sense, namely, the strategy is also independent of the call stack. This fails badly for multi-exit RMDPs, as we will see. We are interested in the following questions:

(1) The *qualitative* termination problem: Is $\mathbf{q}^*_{(u,ex)} = 1$?
(2) The *quantitative* termination problems: Given $r \in [0,1]$, is $q^*_{(u,ex)} \geq r$? Is $q^*_{(u,ex)} = r$? Or we may wish to compute or approximate probabilities $q^*_{(u,ex)}$.

More generally, we can ask model checking questions, where, given a Σ-labeling of vertices, and e.g., an LTL formula φ over Σ, we ask what is the supremum probability with which player 1 can force the satisfaction of property φ? We refrain from formal definitions due to space (see,e.g., [CY98,EY05b]). Our results for model checking will be negative: undecidability, stemming from the undecidability of termination problems for general RMDPs.

3 Systems of nonlinear min-max equations for 1-exit RSSGs

We generalize the monotone nonlinear system of equations for RMCs ([EY05a]) to monotone nonlinear min-max systems for 1-exit RSSGs, whose Least Fixed Point yields the desired probabilities $q^*_{(u,ex)}$. Let us use a variable $x_{(u,ex)}$ for each unknown $q^*_{(u,ex)}$. We will often find it convenient to index the variables $x_{(u,ex)}$ according to a fixed order (say lexicographical), 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. In this way we obtain a vector of variables: $\mathbf{x} = (x_1\ x_2 \ldots x_n)^T$.

Definition 1. *Given 1-exit RSSG $A = (A_1, \ldots, A_k)$, we define a system of polynomial/min-max 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 of the form $x_{(u,ex)} = P_{(u,ex)}(\mathbf{x})$, for each variable $x_{(u,ex)}$. There are 5 cases to distinguish, based on what "type" of vertex u is:*

1. *Type I: $u = ex$. In this case: $x_{(ex,ex)} = 1$.*
2. *Type II: $\texttt{pl}(u) = 0$ & $u \in (N_i \backslash \{ex\}) \cup \mathrm{Return}^i$: $x_{(u,ex)} = \sum_{\{v \mid (u,p_{u,v},v) \in \delta\}} p_{u,v} x_{(v,ex)}$.*
   (If u has no outgoing transitions, this equation is by definition $x_{(u,ex)} = 0$.)
3. *Type III: $u = (b, en)$ is a call port: $x_{((b,en),ex)} = x_{(en,ex')} \cdot x_{((b,ex'),ex)}$, where $ex' \in Ex_{Y(b)}$ is the unique exit of $A_{Y(b)}$.*
4. *Type IV: $\texttt{pl}(u) = 1$ & $u \in (N_i \backslash \{ex\}) \cup \mathrm{Return}^i$: $x_{(u,ex)} = \max_{\{v \mid (u,\perp,v) \in \delta\}} x_{(v,ex)}$.*
   (If u has no outgoing transitions, we define $\max(\emptyset) = 0$.)
5. *Type V: $\texttt{pl}(u) = 2$ and $u \in (N_i \backslash \{ex\}) \cup \mathrm{Return}^i$: $x_{(u,ex)} = \min_{\{v \mid (u,\perp,v) \in \delta\}} x_{(v,ex)}$.*
   (If u has no outgoing transitions, we define $\min(\emptyset) = 0$.)

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 1-exit RSSG A, we can easily construct $\mathbf{x} = P(\mathbf{x})$ in linear time. We now identify a particular solution to $\mathbf{x} = P(\mathbf{x})$, called the *Least Fixed Point* (LFP) solution, which gives precisely the termination game values. For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, define the partial-order $\mathbf{x} \preceq \mathbf{y}$ to mean $x_j \leq y_j$ for every coordinate j. For $D \subseteq \mathbb{R}^n$, we 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 define $P^k(\mathbf{x}) = P(P^{k-1}(\mathbf{x}))$, for $k > 1$. Let $\mathbf{q}^* \in \mathbb{R}^n$ denote the n-vector $q^*_{(u,ex)}$ (using the same indexing as used for $\mathbf{x}$). For $k \geq 0$, let $\mathbf{q}^k$ denote, similarly, the n-vector $q^k_{(u,ex)}$. Let $\mathbf{0}$ ($\mathbf{1}$) denote the n-vector consisting of 0 (respectively, 1) in every coordinate. Define $\mathbf{x}^0 = \mathbf{0}$, and for $k \geq 1$, define $\mathbf{x}^k = P(\mathbf{x}^{k-1}) = P^k(\mathbf{0})$.

Theorem 1. *Let $\mathbf{x} = P(\mathbf{x})$ be the system S_A associated with 1-exit RSSG A.*

1. *$P : \mathbb{R}^n \mapsto \mathbb{R}^n$ is monotone on $\mathbb{R}^n_{\geq 0}$. Hence, for $k \geq 0$, $\mathbf{0} \preceq \mathbf{x}^k \preceq \mathbf{x}^{k+1}$.*
2. *For all $k \geq 0$, $\mathbf{q}^k \preceq \mathbf{x}^{k+1} \preceq \mathbf{q}^{2^k}$.*
3. *$\mathbf{q}^* = P(\mathbf{q}^*)$. In other words, $\mathbf{q}^*$ is a* fixed point *of the map P.*
4. *For all $k \geq 0$, $\mathbf{x}^k \preceq \mathbf{q}^*$.*
5. *For all $\mathbf{q}' \in \mathbb{R}^n_{\geq 0}$, if $\mathbf{q}' = P(\mathbf{q}')$, then $\mathbf{q}^* \preceq \mathbf{q}'$.*
   In other words, $\mathbf{q}^*$ is the Least Fixed Point, $\mathrm{LFP}(P)$, *of $P : \mathbb{R}^n_{\geq 0} \mapsto \mathbb{R}^n_{\geq 0}$.*
6. *$\mathbf{q}^* = \lim_{k \to \infty} \mathbf{x}^k = \lim_{k \to \infty} \mathbf{q}^k$.*

The proofs are omitted due to space. They are similar to those of an analogous theorem in [EY05a] for nonlinear systems associated with RMCs, but some parts are substantially more tricky because of the players. We sketch here the idea for part (5). Consider any fixpoint $\mathbf{q}'$ of the equations, i.e., $\mathbf{q}' = P(\mathbf{q}')$. Let τ' be the (S&M) strategy for player 2 that always picks, at any state $\langle \beta, u \rangle$, for vertex $u \in \texttt{pl}^{-1}(2)$, the particular successor v of u such that $v = \arg\min_{\{v \mid (u,\perp,v) \in \delta\}} \mathbf{q}'_{(v,ex)}$

(breaking ties, say, lexicographically). Then we prove a lemma stating that, for all strategies $\sigma \in \Psi_1$ of player 1, and for all $k \geq 0$, $\mathbf{q}^{k,\sigma,\tau'} \preceq \mathbf{q}'$. The lemma implies that $\mathbf{q}^{*,\sigma,\tau'} = \lim_{k\to\infty} \mathbf{q}^{k,\sigma,\tau'} \preceq \mathbf{q}'$. This holds for any strategy $\sigma \in \Psi_1$. Therefore, $\sup_{\sigma\in\Psi_1} q^{*,\sigma,\tau'}_{(u,ex)} \leq \mathbf{q}'_{(u,ex)}$, for every vertex u. Thus, by the determinacy of RSSG games, it follows that $\mathbf{q}^*_{(u,ex)} = \inf_{\tau\in\Psi_2} \sup_{\sigma\in\Psi_1} q^{*,\sigma,\tau}_{(u,ex)} \leq \sup_{\sigma\in\Psi_1} q^{*,\sigma,\tau'}_{(u,ex)} \leq \mathbf{q}'_{(u,ex)}$, for all vertices u. In other words, $\mathbf{q}^* \preceq \mathbf{q}'$. □

4 S&M Determinacy

We now identify a very restricted kind of strategy that suffices as an optimal strategy in 1-exit RSSGs. Call a strategy *Stackless & Memoryless (S&M)* if it is not only independent of the history of the game, but also independent of the current call stack, i.e., only depends on the current vertex. (See also [ATM03], where such strategies are called *modular* strategies.)

Corollary 1. *In every 1-exit RSSG termination game, player 2 (the minimizer) has an optimal S&M strategy.*

Proof. Consider the strategy τ' in the proof of part (5) of Theorem 1, chosen not for just any fixed point $\mathbf{q}'$, but for $\mathbf{q}^*$ itself. □

Far less trivially, we establish next that player 1 (the maximizer) also has an optimal S&M strategy and thus the game is *S&M-determined*, meaning both players have optimal S&M strategies. (Note that the game is not symmetric with respect to the two players.)

Theorem 2. *Every 1-exit RSSG termination game is S&M-determined.*

Although the statement is intuitive, the proof is quite nontrivial and delicate; the full proof is given in the full paper. We sketch the approach here. By Corollary 1, we only need to show that player 1 has an optimal S&M strategy. Let σ be any S&M strategy for player 1, and let $\mathbf{q}^{*,\sigma} = \inf_{\tau\in\Psi_2} \mathbf{q}^{*,\sigma,\tau}$. If $\mathbf{q}^{*,\sigma}$ is a fixpoint of the equations then it follows that it is the least fixpoint and σ is optimal. On the other hand, it can be shown that $\mathbf{q}^{*,\sigma}$ satisfies all the equations except possibly for some type IV equations. We argue that if u is such a vertex (belonging to player 1) whose equation is violated, then switching to another strategy σ' where u picks another successor leads to a strictly better strategy than σ (for any strategy of player 2). This is the heart of the proof. We parameterize the game with respect to the value t at vertex u, and we express the optimal values of the other vertices z (for all strategies τ of player 2) as functions $f_z(t)$. We then carefully analyze the properties of these functions, which are power series in t with non-negative coefficients, and we analyze their fixpoints, and conclude that switching the choice at vertex u leads to a strategy σ' that has at least as great value as σ at every vertex, and strictly better at u. We repeat the process until we arrive at a S&M strategy σ^* whose probabilities satisfy all the equations, and hence it is optimal. We refer to the full paper for the details. □

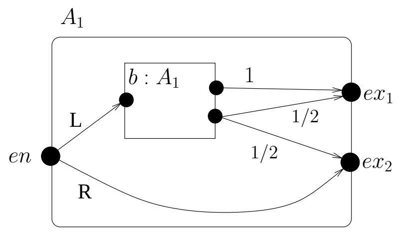

Fig. 1. *2-exit RMDPs:* no optimal strategy exists for terminating at ex_1.

Already for 2-exit RMDP termination, not only are there no optimal S&M strategies for player 1, there are in general no optimal strategies at all! Figure 1 illustrates this. In this 2-exit maximizing RMDP the *supremum* probability of terminating at exit ex_1 starting from en is 1. However, no strategy player 1 achieves this. Specifically, for $n \geq 0$, the strategy $L^n R$ terminates at ex_1 with probability $(1 - \frac{1}{2^n})$. Note that any S&M strategy for player 1 would yield probability 0 of terminating at ex_1, so such strategies are the worst possible.

5 Termination problems for 1-exit RMDPs & RSSGs

Using Corollary 1 and Theorem 2, and results from [EY05a], we can show the following results for qualitative termination of 1-exit RMDPs and 1-exit RSSGs:

Theorem 3.

1. *We can decide in P-time if the value of a 1-exit RSSG termination game (and optimal termination probability in a maximizing or minimizing 1-exit RMDP) is exactly 0.*
2. *We can decide in NP whether the maximum probability of termination in a maximizing 1-exit RMDP is exactly* 1*, and in coNP whether the minimum probability of termination in a minimizing 1-exit RMDP is exactly* 1*.*
3. *Deciding whether a 1-exit RSSG termination game has value 1 is in* $\Sigma_2^P \cap \Pi_2^P$*.*
4. *For 1-exit* `lr`*-RMDPs we can compute the exact optimal (rational) termination probability in P-time, and for 1-exit* `lr`*-RSSGs we can compute the exact optimal (rational) value of the termination game in NP∩co-NP.*

Part (1) is done via a fixpoint algorithm; parts (2) and (3) involve guessing the optimal S&M strategies and verifying the optimality with the appropriate complexity using techniques from [EY05a]; part (4) exploits the fact that for 1-exit `lr`-RMDPs & `lr`-RSSGs the non-linear system $\mathbf{x} = P(\mathbf{x})$ can be decomposed into linear parts that can be solved sequentially by linear programming (and by guessing strategies for RSSGs); see the full paper. □

We next show that quantitative termination questions for 1-exit RMDPs and 1-exit RSSGs can be answered in PSPACE by appealing to the deep algorithms for deciding the *Existential Theory of Reals*, ***ExTh***($\mathbb{R}$). A first-order sentence in the theory of reals is formed from quantifiers and boolean connectives over a vocabulary with "atomic predicates" of the form: $f_i(\mathbf{x})\Delta 0$, where the f_i, are multi-variate polynomials with rational coefficients over the variables $\mathbf{x} = x_1, \ldots, x_n$, and Δ is any comparison operator: $=, <$, or $\leq$. The *existential theory of reals*, $\boldsymbol{ExTh}(\mathbb{R})$, consists of prenex sentences: $\exists x_1, \ldots, x_n R(x_1, \ldots, x_n)$, where R is a boolean combination of "atomic predicates". Beginning with Tarski, algorithms for deciding the theory of reals and fragments such as $\boldsymbol{ExTh}(\mathbb{R})$ have been deeply investigated. Current, it is known that $\boldsymbol{ExTh}(\mathbb{R})$ can be decided in PSPACE [Can88,Ren92,BPR96]. Furthermore, it can be decided in exponential time where the exponent depends (linearly) only on the number of variables; hence for a fixed number of variables the time is polynomial.

Suppose we want to decide whether a vector $\mathbf{c} = [c_1, \ldots, c_n]^T$ of rational numbers is $LFP(P)$, where $\mathbf{x} = P(\mathbf{x})$ is the system of equations for a given 1-exit RSSG. Consider the sentence:

$$\varphi \equiv \exists x_1, \ldots, x_n \bigwedge_{i=1}^{n} (P_i(x_1, \ldots, x_n) = x_i) \wedge \bigwedge_{i=1}^{n} (x_i = c_i)$$

φ is true iff $\mathbf{c} = P(\mathbf{c})$. For type I, II, and III nodes, P_i is a polynomial. It remains to show how to encode, in arithmetic, the predicate "$P_i(x_1, \ldots, x_n) = x_i$" in the case (IV) where $P_i(x_1, \ldots, x_n) = \max_{j \in J} x_j$, and in the case (V) where $P_i(x_1, \ldots, x_n) = \min_{j \in J} x_j$, for some subset $J \subseteq \{1, \ldots, n\}$. For type IV nodes, note that $x_i = \max_{j \in J} x_j$ iff $\bigwedge_{j \in J} x_i \geq x_j \wedge (\bigvee_{j \in J} x_i \leq x_j)$. Likewise, for type V nodes, $x_i = \min_{j \in J} x_j$ iff $\bigwedge_{j \in J} x_i \leq x_j \wedge (\bigvee_{j \in J} x_i \geq x_j)$. Thus, we can encode the predicates $x_i = P_i(x_1, \ldots, x_n)$ as a boolean combination of quantifier-free predicates, and we can encode the sentence φ in $\boldsymbol{ExTh}(\mathbb{R})$. To guarantee that $\mathbf{c} = \mathrm{LFP}(P)$, we need in addition to check the following sentence:

$$\psi \equiv \exists x_1, \ldots, x_n \bigwedge_{i=1}^{n} (P_i(x_1, \ldots, x_n) = x_i) \wedge \bigwedge_{i=1}^{n} (0 \leq x_i) \wedge \bigvee_{i=1}^{n} (x_i < c_i)$$

ψ is false iff there is no solution $\mathbf{z} \in \mathbb{R}^n_{\geq 0}$ to $\mathbf{x} = P(\mathbf{x})$ such that $\mathbf{c} \not\leq \mathbf{z}$. Hence, to decide whether $\mathbf{c} = \mathrm{LFP}(P)$, we only need two queries to $\boldsymbol{ExTh}(\mathbb{R})$. Namely, we check that φ is true, and hence $\mathbf{c} = P(\mathbf{c})$, and that ψ is false, and hence $\mathbf{c} = \mathrm{LFP}(P)$. If we only want to check an inequality $\mathbf{q}^*_j \leq c_j$, then let φ' be φ with the last conjunction of equations replaced by $\bigwedge_{i=1}^{n} (0 \leq x_i) \wedge (x_j \leq c_j)$. Applying the results on $\boldsymbol{ExTh}(\mathbb{R})$, we obtain the following:

Theorem 4. *Given a 1-exit RSSG A and a vector of rational probabilities $\mathbf{c}$, there is a PSPACE algorithm to decide whether $\mathbf{q}^* = \mathbf{c}$, as well as to decide whether $\mathbf{q}^*_j \Delta c_j$, for any comparison operator Δ. Moreover, the running time is $O(|A|^{O(1)} \cdot 2^{O(n)})$ where n is the number of variables in $\mathbf{x} = P(\mathbf{x})$. Hence the running time is polynomial if n is bounded.*

Since $\mathbf{0} \leq \mathrm{LFP}(P) \leq \mathbf{1}$, we can use such queries to $\boldsymbol{ExTh}(\mathbb{R})$ in a "binary search" to "narrow in" on the value of each coordinate of $\mathrm{LFP}(P)$. Via obvious modifications of sentences like ψ, we can gain one extra bit of precision on the exact value of each c_i with one extra query to $\boldsymbol{ExTh}(\mathbb{R})$. This yields:

Theorem 5. *Given 1-exit RSSG A and a number j in unary, there is an algorithm that approximates every coordinate of* $\mathbf{q}^*$ *to within j bits of precision in PSPACE. Moreover, the running time is* $O(j \cdot |A|^{O(1)} \cdot 2^{O(n)})$*, where n is the number of variables in* $\mathbf{x} = P(\mathbf{x})$.

6 Multi-exit RMDP termination and 1-exit RMDP model checking: Undecidability

We next show strong undecidability results for RMDPs, and thus for RSSGs.

Theorem 6. *Given a multi-exit linearly-recursive RMDP, A, entry en and exit ex, it is undecidable whether* $q^*_{(en,ex)} = 1$*. This is so even when the number of exits in each component of A is bounded by a fixed constant. Furthermore, there is no algorithm that approximates the probability* $q^*_{(en,ex)}$ *within any constant (multiplicative) factor. In particular:*
1. For any fixed rational ϵ *with* $0 < \epsilon < 1$*, given* `lr`*-RMDP A with only one component such that either* $q^*_{(en,ex)} > 1 - \epsilon$ *or* $q^*_{(en,ex)} < \epsilon$*, it is undecidable to distinguish which is the case.*
2. For any fixed rational ϵ *with* $0 < \epsilon < 1$*, given a* `lr`*-RMDP A with only two components such that either* $q^*_{(en,ex)} = 1$ *or* $q^*_{(en,ex)} < \epsilon$*, it is undecidable to distinguish which is the case.*

We have two proofs: one is a reduction from the halting problem for 2-counter machines. The second is a reduction from the emptiness problem for Probabilistic Finite Automata (PFAs). The latter reduction is simpler and connects RMDPs to the well-studied area of PFAs, allowing us to leverage extensive research in that area [Paz71,CL89,BC03]. We show that PFAs are, in effect, a special case of RMDPs. Recall, a PFA is a FA whose transitions from a state on a given input are probabilistic. The PFA emptiness problem is to decide for a given PFA A and threshold λ, whether there is a word accepted by A with probability $> \lambda$. It is undecidable in strong ways ([Paz71,CL89,BC03]). Our reduction constructs from a PFA A a `lr`-RMDP whose termination probability under the optimal strategy is precisely the supremum probability of acceptance of any word by A. Please see the full paper for details. With a modified construction we show:

Theorem 7. *Qualitative & quantitative LTL model checking for 1-exit* `lr`*-RMDPs is undecidable; this holds even for a fixed property; moreover, the optimum probability of the property can't be approximated within any constant factor.*

Acknowledgement Thanks to Amir Pnueli for asking us about `lr`-RMCs.

References

[AEY01] R. Alur, K. Etessami, and M. Yannakakis. Analysis of recursive state machines. In *Proc. of 13th CAV*, pages 304–313, 2001.
[ATM03] R. Alur, S. La Torre, and P. Madhusudan. Modular strategies for recursive game graphs. In *TACAS*, volume 2619 of *LNCS*, pages 363–378, 2003.

[BC03] V. Blondel and V. Canterini. Undecidable problems for probabilistic automata of fixed dimension. *Theory of Comput. Sys.*, 36:231–245, 2003.

[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. 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.

[CL89] A. Condon and R. Lipton. The complexity of space bounded interactive proofs. In *Proc. of 30th IEEE FOCS*, 1989.

[Con92] A. Condon. The complexity of stochastic games. *Inf. & Comp.*, 96(2):203–224, 1992.

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

[CY98] C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. *IEEE Trans. on Automatic Control*, 43(10):1399–1418, 1998.

[dAKN+00] L. de Alfaro, M. Kwiatkowska, G. Norman, D. Parker, and R. Segala. Symbolic model checking of probabilistic processes using MTBDDs and the kronecker representation. In *Proc. of 6th TACAS*, pages 395–410, 2000.

[EKM04] J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. In *Proc. of 19th IEEE LICS'04*, 2004.

[Ete04] K. Etessami. Analysis of recursive game graphs using data flow equations. In *Proc. 5th VMCAI*, vol. 2937 *LNCS*, pages 282–296. Springer, 2004.

[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. (Tech. Report, U. Edinburgh, June 2004).

[EY05b] K. Etessami and M. Yannakakis. Algorithmic verification of recursive probabilistic state machines. In *Proc. 11th TACAS*, vol. 3440 of LNCS, 2005.

[FS02] E. Feinberg and A. Shwartz, editors. *Handbook of Markov Decision Processes*. Kluwer, 2002.

[FV97] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.

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

[HSP83] S. Hart, M. Sharir, and A. Pnueli. Termination of probabilistic concurrent programs. *ACM Trans. Prog. Lang. & Sys.*, 5(3):356–380, 1983.

[Jag75] P. Jagers. *Branching Processes with Biological Applications*. Wiley, 1975.

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

[Mar98] D. A. Martin. Determinacy of Blackwell games. *J. Symb. Logic*, 63(4):1565–1581, 1998.

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

[Paz71] A. Paz. *Introduction to Probabilistic Automata*. Academic Press, 1971.

[Put94] M. L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[Ren92] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals, parts I-III. *J. Symb. Comp.*, 13(3):255–352, 1992.

[SBH+94] Y. Sakakibara, M. Brown, R Hughey, I.S. Mian, K. Sjolander, R. Underwood, and D. Haussler. Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Research*, 22(23):5112–5120, 1994.

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